12-1-1998

AUGMENTED TRANSITION NETWORKS AS SEMANTIC MODELS FOR MULTIbAEDIA PRESENTATIONS, MULTIMEDIA DATABASE SEAR.CHING, AND MULTIMEDIA BROWSING

Shu-Ching Chen
Purdue University School of Electrical and Computer Engineering

R. L. Kashyap
Purdue University School of Electrical and Computer Engineering

AUGMENTED TRANSITION NETWORKS AS SEMANTIC MODELS FOR MULTIMEDIA PRESENTATIONS, MULTIMEDIA DATABASE SEARCHING, AND MULTIMEDIA BROWSING

SHU-CHING CHEN
R. L. KASHYAP

TR-ECE 98-15
DECEMBER 1998

SCHOOL OF ELECTRICAL
AND COMPUTER ENGINEERING
PURDUE UNIVERSITY
WEST LAFAYETTE, INDIANA 47907-1285

AUGMENTED TRANSITION NETWORKS AS SEMANTIC MODELS FOR MULTIMEDIA PRESENTATIONS, MULTIMEDIA DATABASE SEARCHING, AND MULTIMEDIA BROWSING [1]

Shu-Ching Chen and R. L. Kashyap

School of Electrical & Computer Engineering
1285 Electrical Engineering Building
Purdue University
West Lafayette, IN 47907–1285

'This work has been partially supported by National Science Foundation under contract IRI 9619812.

TABLE OF CONTENTS

Page

Page

ABSTRACT

As more information sources become available in multimedia systems, the development of abstract semantic models for video, audio, text, and image data becomes very important. An abstract semantic model has two requirements. First, it should be rich enough to provide a friendly interface of multimedia presentation synchronization schedules to the users. Second, it should be a good programming data structure for implementation to control multimedia playback.

An abstract semantic model based on an augmented transition network (ATN) is presented. The inputs for ATNs are modeled by multimedia input strings. Multimedia input strings provide an efficient means for iconic indexing of the temporal/spatial relations of media streams and semantic objects. An ATN and its subnetworks are used to represent the appearing sequence of media streams and semantic objects. The arc libel is a substring of a multimedia input string. In this design, a presentation is driven by a multimedia input string. Each subnetwork has its own multimedia input string. Database queries relative to text, image, and video can be answered via substring matching at subnetworks. Subnetworks also can be some existing multimedia presentations to be embedded in other presentations to make module design possible in a multimedia authoring environment. The conditions are checked to see whether certain criteria are satisfied. If they are, a set of corresponding actions are activated. Multimedia browsing allows users the flexibility to select any part of the presentation they prefer to see. This means that an ATN and its subnetworks can be included in a multimedia database system which is controlled by a database management system (DBMS). User interactions and loops are also provided in an ATN. Therefore, ATNs provide three major capabilities: multimedia presentations, temporal/spatial multimedia database searching, and multimedia browsing.

Keywords: Multimedia Presentations, Multimedia Database Systems, Augmented Transition Network (ATN), Multimedia Input Strings.

1. INTRODUCTION

1.1 Motivation and Problem Definition

In Multimedia systems, a variety of information sources – text, voice, image, audio, animation, and video – are delivered synchronously or asynchronously via more than one device. The important characteristic of such a system is that all of the different media are brought together into one single unit, all controlled by a computer. Normally, multimedia systems require the management and delivery of extremely large bodies of data at very high rates and may require the delivery with real-time constraints. In traditional database management systems (DBMS), such as relational database systems, only text information is stored in the database and there is no need to consider the synchronicity among media. In object-oriented database systems, a database may include image data, and the DBMS still is not designed to support multimedia information. Multimedia extension is needed to handle the mismatch between multimedia data and the conventional object-oriented database management systems (Chen, et al., 1995). In multimedia database systems, a new design of multimedia database management systems (MDBMS) is required to handle the temporal and spatial requirements, and the rich semantics of multimedia data such as text, image, audio, and video. The temporal requirements are that media need to be synchronous and to be presented at the specified time that was given at authoring time. The spatial requirement is that the DBMS needs to handle the layout of the media at a certain point in time. For image and video frames, the DBMS needs to keep the relative positions of semantic objects (building, car, etc.) so that users can issue queries, such as, "Find a video clip that has one car in front of a building." In order to keep the rich semantic information, abstract semantic models are developed to let users specify the temporal and spatial requirements at the time of authoring the objects, and to store a great deal of useful information (such as video clip start/end time, start/end frame number, and semantic objects relative spatial

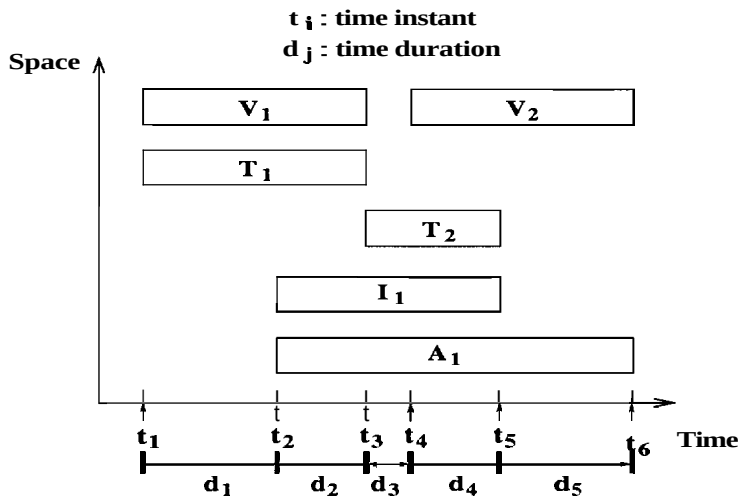


Fig. 1.1. Timeline for Multimedia Presentation. t_1 to t_6 are the time instances. d_1 is time duration between t_1 and t_2 and so on.

locations). Also, the semantic model can model the hierarchy of visual contents so that users can browse and decide on various scenarios they want to see. Therefore a, semantic model should provide presentation, database searching, and browsing capabilities to users. Some researchers have proposed semantic models such as petri-net models, tinne-interval based models, graphic models, and timeline models. However, most of these models can deal with only certain parts of the requirements for the MDBMS.

For a multimedia database system to be more intelligent, more flexible, and more efficient (for real time response) than existing methods, the knowledge embedded in images or videos, especially spatial knowledge, should be captured by the data structure as much as possible. Extracting; information from images/videos is time consuming. In order to provide fast response for real time applications, information or knowledge needs to be extracted from images/videos item by item in advance and stored for later retrieval. For example, to do spatial reasoning we would have to store numerous spatial relations among; objects (Chang, 1988). The semantic model proposed in this paper belongs to this category which will do the preprocessing for image/videos and store in the databases for later retrieval.

1.2 Existing Semantic Models

We have mentioned the importance of abstract semantic models. Many semantic models have been proposed to model temporal and spatial relations (Blakowski and Huebel, 1991;

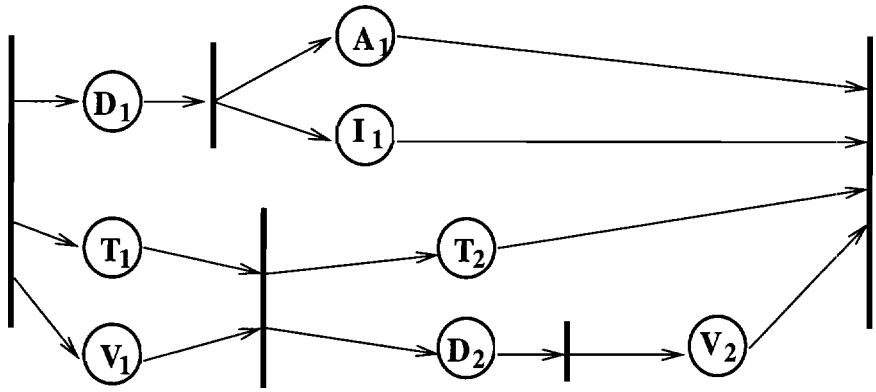


Fig. 1.2. An OCPN example for Figure 1.1: D_1 is the delay for media streams I_1 and A_1 to display. D_2 is the delay for V_2 to display.

Little and Ghafoor, 1990a; Little and Ghafoor, 1990b; Little and Ghafoor, 1993; Hirzalla and Karmouch, 1995; Chang et al., 1995; Lin et al., 1996; Wahl and Rothermel.,1994; Buchanan and Zellweger, 1993ab; Yahya and Chang, 1996).

A media stream in a multimedia presentation is defined as one or more letters subscripted by digit(s) in this paper. The letters A, I, T, and V represent audio, image, text, and video media streams, respectively. The subscript digit(s) is to denote the segment number in the correspondling media stream. For example, V_1 denotes video stream segment 1. Figure 1.1 is a timeline to represent a multimedia presentation. The presentation starts at time t_1 and ends at time t_6. At time t_1, media streams V_1 (Video 1) and T_1 (Text 1) start to play at the same time and continue to play. At time t_2, I_1 (Image 1) and A_1 (Audio 1) begin and overlap with V_1 and T_1. The duration d_1 is the time difference between t_1 and t_2. V_1 and T_1 end at time t_3 which T_2 starts. The process continues till the end of the presentation. The timeline representation can model the temporal relations of media streams in a multimedia presentation (Blakowski and Huebel, 1991). Every presentation needs to strictly follow the prespecified sequence. However, it cannot handle the network delay, buffer limitations, etc.

Little and Ghafoor (1990; 1993) proposed an Object Composition Petri Net (OCPN) model based on the logic of temporal intervals and Timed Petri Nets. OCPN was proposed to store, retrieve, and communicate between multimedia objects. This model is a modification of earlier Petri net models and consists of a set of transitions (bars), a set of places (circles), and a set of directed arcs. In OCPN (as shown in Figure 1.2), each place (circle) contains

the required presentation resource (device), the time required to output the presentation data, and spatial/content information. Each place (circle) is represented by a state node in the OCPN model. The *transitions* (bars) in the net indicate points of synchronization and processing. OCPN is a network model and a good data structure for controlling the synchronization of the multimedia presentation. A network model can easily show the time flow of a presentation. Therefore, OCPN can serve as a visualization structure for users to understand the presentation sequence of media streams.

Figure 1.2 shows the OCPN for the same example in Figure 1.1. Media streams V_1 and T_1 start to display and I_1 and A_1 join to display after a duration of delay which is denoted by D_1. After V_1 and T_1 finish, T_2 together with I_1 and A_1 to display. Then V_2 join to display after duration D_2. OCPN can handle the synchronization and quality-of-service (QoS) for real-time multimedia presentations. Many later abstract semantic models are based on petri-net or OCPN (Chang et al., 1995; Lin et al.. 1996; Yahya and Chang, 1996; Thimm and Klass, 1996).

1.3 Shortcomings of Existing Semantic Models

This section delineates the disadvantages of existing semantic models that are addressed in this paper:

- User interactions are not included in the conceptual models proposed by (Little and Ghafoor, 1990a; Chang et al., 1995; Lin et al., 1996; Yahya and C'hang, 1996). In interactive multimedia presentations, users should have the flexibility to decide various scenarios they want to see. This means that two-way communications should be captured by the conceptual model. Figures 1.3 and 1.4 are two timelines to model user selections. In Figure 1.3, V_1 (Video 1) and T_1 (Text 1) are displayed beginning at time t_1 and ending at time t_2. Then the presentation provides two selection buttons to let users make the selection. If B_1 is chosen, V_2 and T_2 are displayed from time t_3 through time t_4. If B_2 is chosen, V_3 and T_3 are displayed with the same duration as V_2 and T_2. At time t_4, both selection paths merge and begin to display V_4 and T_4. This means no matter B_1 or B_2 is chosen, the presentation will share the same presentation sequence after time t_4. Figure 1.4 represents the same scenario as Figure 1.3 by using

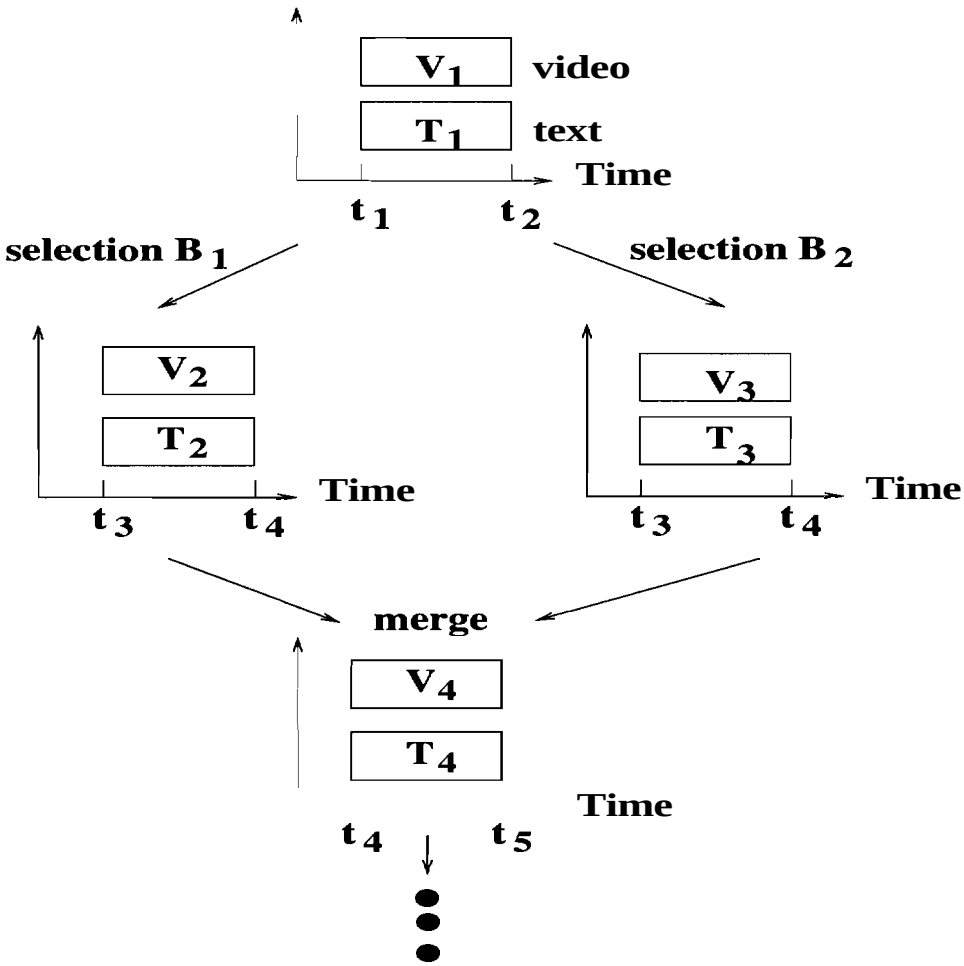


Fig. 1.3. Timeline model with commands in user selections.

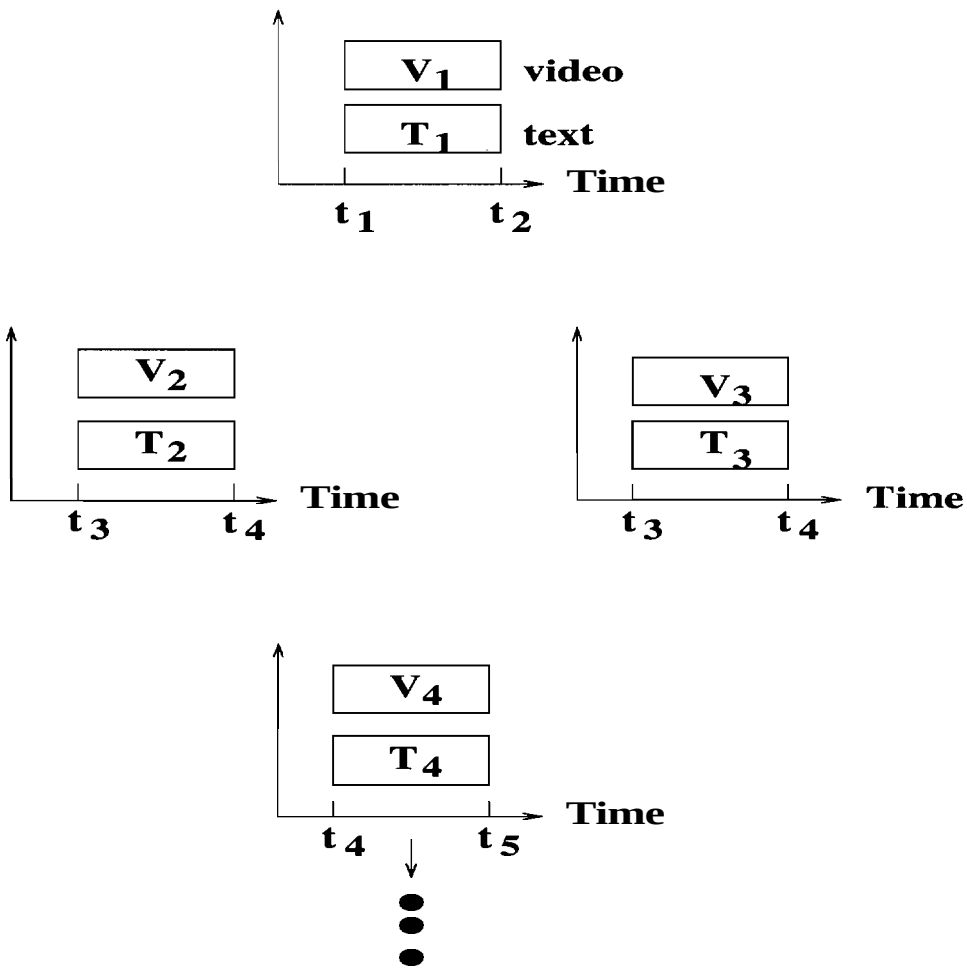


Fig. 1.4. Timeline model with no commands in user selections.

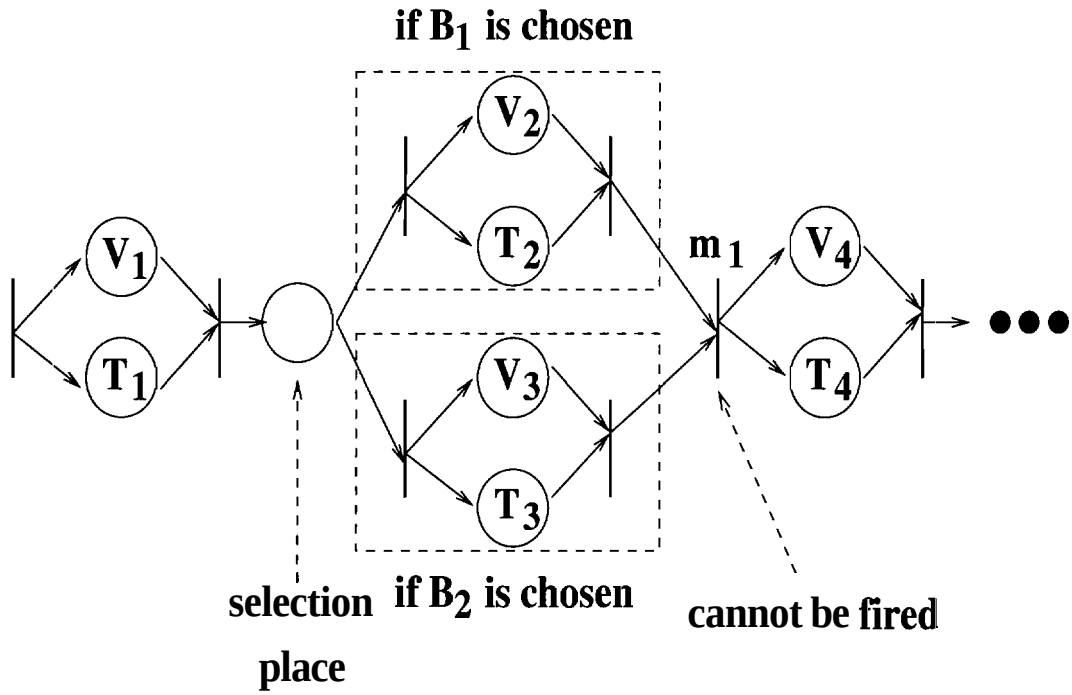


Fig. 1.5. OCPN with user selections: m_1 cannot be fired since one of m_1's incoming arcs is active and the other incoming arc is not active. Different selection paths cannot merge together in OCPN.

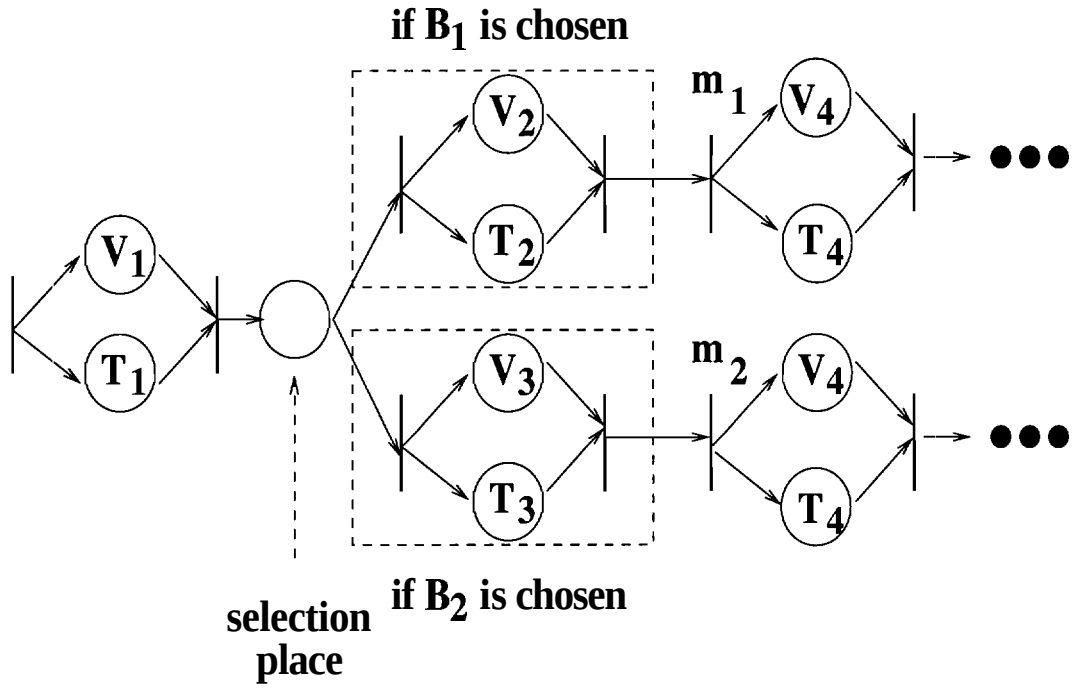


Fig. 1.6. In order to allow user selections in OCPN, transition m_1 and m_2 are used for different selection paths. V_4 and T_4 need to present in the two selection paths.

only timeline. As shown in Figure 1.4, it is difficult to let users understand the selection sequences without additional commands and arcs. OCPN is a form of *marked graph* so that each place in an OCPN has exactly one incoming arc and one outgoing arc. *Marked graph* can only model those systems whose control flow has no branch. Hence, it can model the parallel activities but not alternative activities;(Peterson, 1977) such as user interactions. In order to model user interactions, Day (1996) proposed an OCF'N-S with user selection functions supplemented in OCPN. In OCPN-S, a selection place which may have more than one outgoing arc to model the alternative activities is defined. Two paths belonging to different selections cannot merge together to avoid the situation as shown in Figure 1.5(a). Figure 1.5(a) represents the same scenario as shown in Figure 1.3. In Figure 1.5(a), after V_1 and T_1 are displayed and the control reaches the selection place. B_1 or B_2 can be chosen so that different media streams will be displayed. Two selection paths merge at transition m_1. Under this situation, transition m_1 cannot be fired. The reason is that the selection path for either B_1 or B_2 can have a token since those places belonging to the unselected paths are excluded from the presentation. In order to solve this problem, OCPN-S need to have different transitions for different selection paths as shown in Figure 1.6. Under this change, media streams V_4 and T_4 need to be duplicated at each presentation sequence. The numbers of nodes and arcs may grow exponentially if a lot of user selections happen in the latter presentation. In applications such as computer-aided instruction (CAI), a situation as in Figure 1.3(b) can happen since CAI applications may involve extensive two-way communications and different selections will merge together occasionally. On the other hand, in ATNs, based on the selections of users, the corresponding input symbol is read, and the conditions and actions are used to pass the control to the desired state to let the presentation continue. Therefore, ATN allows nested selections and merges for the scenario in Figure 1.3(b). In addition, the shared presentation in Figure 1.3(b) can be modeled by using embedded presentation in ATNs as shown in section 5.3.

- Although the presentation deadline for each media stream may be decided at the authoring time, the conceptual model should have the ability to handle the communication delays for real-time presentations to maintain the quality of service (QoS). Different kinds of delays may need disparate strategies to maintain QoS of the presentations. Therefore, the conceptual model should have the mechanisms to handle the different delay situations. The model proposed by (Hirzalla et al., 1995) handles user interaction delays but it does not handle other communication delays. Some models handle a communication delay by adjusting the playout deadline schedule for the media streams; however they do not provide necessary actions for different communication delays (Little and Ghafoor, 1990a; Little and Ghafoor, 1990b; Little and Ghafoor, 1993; Hirzalla et al., 1995; Chang et al., 1995; Lin et al., 1996). The multimedia transition table in the ATN allows users to specify different actions for different communication delays and uncertainty of the users' response time when two way communication occurs.

- The conceptual model needs to serve as the graphical interface for users to understand the whole presentation schedule and the appearance of different media streams at different times. It also serves as a data structure for presentation control. Existing conceptual models are either too complex for the users to understand (Little and Ghafoor,1990a; Little and Ghafoor,1990b; Little and Ghafoor,1993; Chang et al.,1995; Lin et al.,1996; Wahl and Rothermel,1994; Yahya and Chang,1996) or too simple to let users see the whole view of the presentation schedule (Blakowski and Huebel, 1991; Hirzalla et al., 1995). The detailed information is placed in the multimedia transition table so that the ATN graph looks more concise and still preserves rich semantic information that is associated with it.

- Few existing conceptual models model both temporal and spatial relations. They either develop a temporal model to capture synchronization information (Blakowski and Huebel, 1991; Little and Ghafoor, 1990a; Little and Ghafoor, 1990b; Little and Ghafoor, 1993; Hirzalla et al., 1995; Chang et al., 1995; Lin et al., 1996; Wahl and Rothermel, 1994; Buchanan and Zellweger, 1993ab; Yahya and Chang, 1996) or use

image/computer vision techniques to get content-based information in the image or video (Arman et al., 1994; Smoliar and Zhang, 1994; Flickner et al., 1995). ATN and its subnetworks can model the temporal relations of media streams and semantic objects, respectively. Based on multimedia input string, the temporal relations of media streams and semantic objects can be obtained so that the database queries that involve spatial relations of semantic objects can be answered.

This paper addresses the above mentioned weaknesses by using an augmented transition network (ATN) to model the multimedia presentations, multimedia database searching, and multimedia browsing.

1.4 The Augmented Transition Network (ATN)

A multimedia environment should not only display media streams to users but also allow two-way communication between users and the multimedia system. The multimedia environment consists of a multimedia presentation system and a multimedia database system. If a multimedia environment has only a presentation system but not a multimedia database system, then it is like a VCR or a TV without feedback from the user. A multimedia database system allows users to specify queries for information. The information may be relative to text data as well as image or video content. By combining a multimedia presentation and multimedia database system, users can specify queries which reflect what they want to see or know. A semantic model that models the presentation has the ability to check the features specified by users in the queries, and maintains the synchronization and quality-of-service (QoS) desired.

A finite state machine (FSM) consists of a network of nodes and directed arcs connecting them. The FSM is a simple transition network. Every language that can be described by an FSM can be described by a regular grammar, and vice versa. The nodes correspond to states and the arcs represent the transitions from state to state. Each arc is labeled with a symbol whose input can cause a transition from the state at the tail of the arc to the state at its head. This feature makes FSM have the ability to model a presentation frorn the initial state to some final states or to let users watch the presentation fast forward or reverse. However, users may want to watch part of a presentation by specifying some features relative to image

or video contents prior to a multimedia presentation, and a designer may want to include other presentations in a presentation. These two features require a pushdown mechanism that permits one to suspend the current process and go to another state to analyze a query that involves temporal, spatial, or spatio-temporal relationships. Since an FSM does not have the mechanism to build up the hierarchical structure; it cannot satisfy these two features.

This weakness can be eliminated by adding a recursive control mechanism to the FSM to form a *recursive transition network* (RTN). A recursive transition network is similar to an FSM with the modifications as follows: all states are given names which are then allowed as part of labels on arcs in addition to the normal input symbols. Based on these labels, subnetworks may be created. Three situations can generate subnetworks. In the first situation, when an input symbol contains an image or a video frame., a subnetwork is generated. A new state is created for the subnetwork if there is any change of the number of semantic objects or any change of the relative position. Therefore, the temporal, spatial, or spatio-temporal relations of the semantic objects are modeled in this subnetwork. In other words, users can choose the scenarios relative to the temporal, spatial, or spatio-temporal relations of the video or image contents that they want to watch via queries. Second, if an input symbol contains a text media stream, the keywords in the text media stream become the input symbols of a subnetwork. A keyword can be a word or a sentence. A new state of the subnetwork is created for each keyword. Keywords are the labels on the arcs. The input symbols of the subnetwork have the same order as the keywords appear in the text. Users can specify the criteria based on a keyword or a combination of keyworcls in the queries. In addition, the information of other databases can be accessed by keywords via the text subnetworks. For example, if a text subnetwork contains the keyword "Purdue University Library," then the Purdue University library database is linked via a query with this keyword. In this design, an ATN can connect multiple existing database systems by passing the control to them. After exiting the linked database system, the control is back to the ATN. Third, if an ATN wants to include another existing presentation (ATN) as a subnetwork, the initial state name of the existing presentation (ATN) is put as the arc label of the ATN. This allows any existing presentations to be embedded in the current ATN to make a new design easier. The advantage is that the other presentation structure is independent of the

current presentation structure. This makes both the designer and users have a clear view of the presentation. Any change in the shared presentation is done in the shared presentation itself. There is no need to modify those presentations which use it as a subnetwork.

Before the control is passed to the subnetwork, the state name at the head of the arc is pushed into the push-down store. The analysis then goes to the subnetwork whose initial state name is part of the arc label. When a final state of the subnetwork is reached, a pop occurs and the control goes back to the state removed from the top of the push-down store.

However, the FSM with recursion cannot describe cross-serial dependencies. For example, network delays may cause some media streams not to be displayed to users at the tentative start time and the preparation time for users to make decisions is unknown when user interactions are provided. In both situations, there is a period of delay which should be propagated to the later presentations. Also, users may specify queries related to semantic objects across several subnetworks. The information in each subnetwork should be kept so that the analysis across multiple subnetworks can be done. For example, the temporal, spatial, or spatio-temporal relations among semantic objects may involve several video subnetworks. The cross-serial dependencies can be obtained by specifying conditions and actions in each arc. The arrangement of states and arcs represents the surface structure of a multimedia presentation sequence. If a user wants to specify a presentation which may be quite different from the surface structure then the actions permit rearrangements and embeddings, and control the synchronization and quality of service of the original presentation sequence. The cross-serial dependencies are achieved by using *variables* and they can be used in later actions or subsequent input symbols to refer to their values. The actions determine additions, subtractions, and changes to the values of *variables* in terms of the current input symbol and conditions. Conditions provide more sensitive controls on the transitions in ATNs. A condition is a combination of checkings involving the feature elements of media streams such as the start time, end time, etc. An action cannot be taken if its condition turns out to be false. Thus more elaborate restrictions can be imposed on the current input symbol for synchronization and quality of service controls. Also, information can be passed along in an ATN to determine future transitions. The recursive FSM with these additions forms an *augmented' recursive transition network* (ATN)

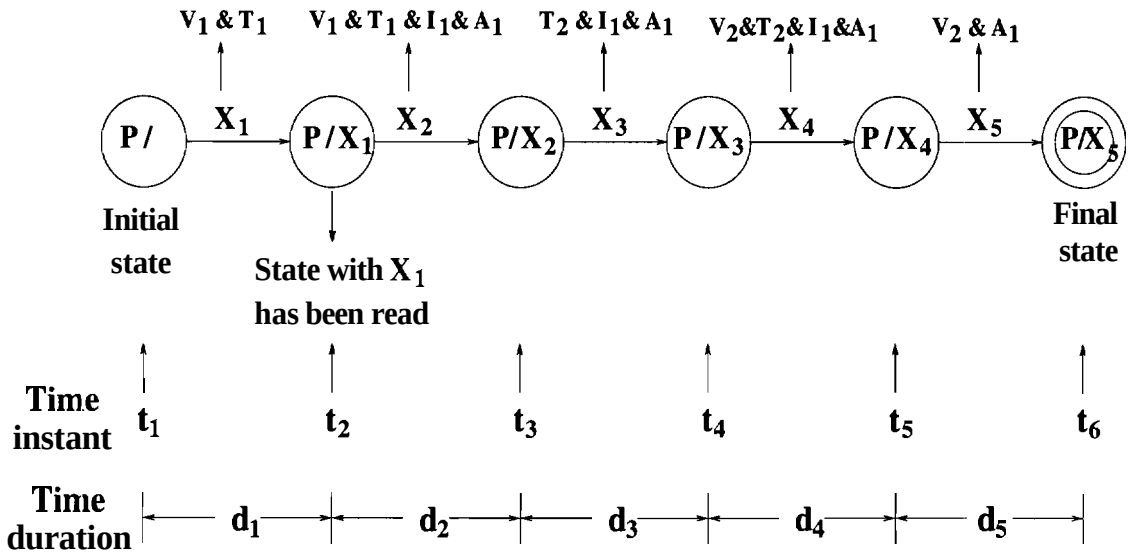


Fig. 1.7. *Transition Network* for Multimedia Presentation.

1.5 Contributions

This paper addresses four important problems:

1. How to model multimedia presentations.
2. How to model temporal and spatial relations of media streams and semantic objects which multimedia database searchings related to different media streams and semantic objects can answer.
3. How to model the video data so that the hierarchy information can 'be preserved and allow users to browse video data.

The following four subsections outline how this paper answers these questions. Section 1.5.5 makes explicit the scope of this work.

The Augmented transition network (ATN), developed by Woods (Woods, 1970), has been used in natural language understanding systems and question answering systems for both text and speech. We use the ATN as a semantic model to model multimedia presentations (Chen and Kashyap, 1997a), multimedia database searching, the temporal, spatial, or spatio-temporal relations of various media streams and semantic objects (Chen and Kashyap, 1997b), and multimedia browsing.

1.5.1 Models for Multimedia **Presentations** Using ATNs and Multimedia Input Strings

To address the first problem (model multimedia presentation), a *transition network* is used to model the presentation sequence (Chen and Kashyap, 1997a). A *transition network* consists of nodes (states) and arcs. Each state has a state name and each arc has an arc label. Each arc label represents the media streams to be displayed in a time duration. Therefore, time intervals can be represented by *transition networks.* In this *transition network,* a new state is created whenever there is any change of media streams in the presentation. There are two situations for the change of media streams and they are as follows::

1. Any media stream finishes to display;

2. Any new media stream joins to display.

Figure 1.7 is a *transition network* for Figure 1.1. There are six states and five arcs which represent six time instants and five time durations, respectively. State names are in the circles to indicate presentation status. State name $P/$ means the beginning of the *transition network* (presentation) and state name P/X_1 denotes the state after X_1 has been read. The reason to use X_i is for convenience purposes. In fact X_1 can be replaced by V_1 and T_1. State name P/X_5 is the final state of the *transition network* to indicate the end of the presentation. State P/X_i represents presentation P just finishs to display X_i and the presentation can proceed without knowing the complete history of the past. There are five occurrences of media stream combinations at each time duration and they are:

1. Duration d_1: V_1 and T_1.

2. Duration d_2: V_1, T_1, I_1, and A_1.

3. Duration d_3: T_2, I_1, and A_1

4. Duration d_4: V_2, T_2, I_1, and A_1.

5. Duration d_5: V_2 and A_1.

Each arc label X_i in Figure *1.4* is created to represent the media stream combination for each duration as above. For example, arc label X_1 represents media streams V_1 and T_1 display

together at duration d_1. A new arc is created when new media streams I_1, and A_1 overlap with V_1 and T_1 to display. A multimedia input string is used as an input for this transition network, and the symbol "&" between media streams indicates these two media streams are display concurrently. A multimedia input string consists of several input symbols and each of them represents the media streams to be displayed at a time interval. The detailed definition of how to use multimedia input string to represent a multimedia presentation will be discussed in Chapter **3.**

1.5.2 Models for Multimedia Database **Searching** Using ATNs and Multimedia Input Strings

In previous subsections, transition networks were used to model multimedia presentations. However, only media streams are included in the transition networks. In order to model spatio-temporal relations of semantic objects, subnetworks are developed to model media streams such as images, video frames, and keywords in texts to form a recursive transition network (RTN). A new state is created in a subnetwork when there is any change in the number of semantic objects or any semantic object spatial location change in the input symbol. For a single image, the subnetwork has only two state nodes and one arc since the number and spatial location of semantic objects will not change.

As mentioned in section 1.4, the hierarchy structure can be constructed by using the subnetworks. The advantage to having subnetworks is that we can separate the coarsed-grain media streams into fine-grained semantic objects. The transition network which contains media streams can provide high level (coarsed-grain) concept to users what kind of media streams are displayed at different durations. The subnetworks can represent low-level (fined-grained) concepts of images, video frames, or texts. If semantic objects; are included in the transition network then it will make this transition network difficult to understand since media streams and semantic objects are mixed together. Multimedia database queries related to images, video frames, or text can be answered by analyzing the corresponding subnetworks (Chen and Kashyap, 1997b). Each subnetwork has a multimedia input string together with it. The multimedia database searching can become a substring matching between the query and the multimedia input string.

In order to let a *recursive transition network* have the ability to control the synchronization and quality of service (QoS), conditions and actions on each arc can handle real-time situations such as network congestion, memory limitation, user interaction delay etc. as mentioned in section 1.4. The details of how to use conditions and actions to maintain the good quality of a multimedia presentation will be discussed in Chapter 3. A *recursive transition network* with conditions and actions on each arc forms an *augmented recursive transition network* (ATN).

There are two primary shortcomings of our work. First, RTN uses either manually identified or image/computer vision techniques to extract information of semantic objects and then generates the subnetworks to model the semantic objects. We assume the image/computer vision or human annotation gives us information of the semantic object's information and do not discuss these issues in this paper. Second, a multimedia input string just models to the semantic object level. Multimedia database queries cannot go beyond this level such as to low-level visual features.

1.5.3 Models for Multimedia **Browsings** Using ATNs and Multimedia Input Strings

Unlike traditional database systems which have text or numerical data, a multimedia database or information system may contain different medias such as text, image, audio, and video. Video is popular in many applications such as education and training, video conferencing, video on demand, news service, and so on. Traditionally, when users want to search a certain content in videos, they need to fast forward or rewind to get a quick overview of interest on the video tape. This is a sequential process and users do not have the chance to choose or jump to specific topic directly. Although disk storage and computer network technologies progress very quickly, the disk storage, network speed, and network bandwidth still cannot meet the requirements of distributed multimedia information applications. How to organize video data and provide the visual content in compact form becomes important in multimedia applications (Yeo and Yeung, 1997). Therefore, users can browse a video sequence clirectly based on their interests so that they can get the necessary information quicker and the amount of data transmission can be reduced. Also, users should have the opportunity to retrieve video materials using database queries. Since video data contains

rich semantic information, database queries should allow users to get high level content such as *scenes* or *shots* and low level content according to the temporal and spatial relations of semantic objects. A semantic object is an object appearing in a video frame such as a "car." Also, a semantic model should have the ability to model visual contents at different granularities so that users can fast browse large video collections.

Many video browsing models are proposed to allow users to visualize video content based on user interactions (Arman et al., 1994; Falchuk and Karmouch, 1995; Flickner et al., 1995; Mills et al., 1992; Oomoto and Tanaka, 1993; Smoliar and Zhang, 1994; Yeo and Yeung, 1997). These models choose representative images using regular time intervals, one image in each shot, all frames with focus key frame at specific place, and so on. Choosing key frames based on regular time intervals may miss some important segments and segments may have multiple key frames with similar contents. One image in each shot also may not capture the temporal and spatial relations of semantic objects. Showing all key frames may let users be confused when too many key frames are displayed at the same time. In addition to using ATNs to model multimedia presentations and multimedia database searching, how to use ATNs and multimedia input strings as video browsing models will be discussed in Chapter 6. Also, key frames selections based on the temporal and spatial relations of semantic objects in each shot will be discussed. The details of how to use a recursive call property in ATNs to model user loops are also presented.

1.5.4 Empirical Studies of Multimedia Semantic Models Based on Different Temporal Relation combinations

Empirical studies of an abstract semantic model, augmented transition network (ATN), with Object Composition Petri Net (OCPN) to model multimedia presentations are performed in this study. The detailed results are presented in Chapter 3. An ATN consists of a set of states and directed arcs and uses a multimedia input string as its input. The advantages to using a multimedia input string are its simplicity and ease of modification. Simulations experiments to compare the nodes, arcs, and transitions between ATN and OCPN based on different numbers of media streams and different combinations of temporal relations are performed in this study. The results show that ATN needs fewer nodes, arcs,

and transitions to represent a multimedia presentation in all cases than OCPN does. These results indicate ATN handles on-line multimedia presentations more efficiently and requires less main memory space.

1.6 Scope of this Research

Automatic segmentation and recognition of semantic objects of medias such as audio and video, and the segmentation of a video clip into different granularities are outside the scope of this paper. We assume that each image or video frame has been segmented automatically or identified manually. The main focus is to model the semantic objects so that the queries related to these semantic objects can be answered quickly. We use ATNs to model the video data so that the browsing or queries related to these video units can be answered quickly. Also, how to use ATNs to share the common video units will be explored too.

1.7 Organization of the Paper

The organization of this proposal is as follows. Chapter 2 surveys the related work in the literature. In Chapter 3, how to use the ATN and the multimedia input strings to model multimedia presentations is discussed. Chapter 4 shows how multimedia database searching can be performed using the ATN and the multimedia input strings. Modeling Embedded presentations, user interactions, and loops using the ATN and the multimedia input strings are presented in Chapter 5. Video browsing using ATN and its subnetworks is discussed in Chapter 6. Conclusions and future work are in Chapter 7.

2. LITERATURE REVIEW

This chapter surveys semantic models in the areas of multimedia presentations, multimedia browsing, and multimedia database systems.

Various types of semantic models have been developed in the past for multimedia presentations, multimedia database searchings, and multimedia browsings. Multimedia semantic models can, based on the underlying paradigm, be classified into the following distinct categories:

(1) petri-net models,

(2) time-interval based models,

(3) graphic; models,

(4) timeline models.

2.1 Petri-Net Models

Little and Ghafoor (1990) proposed an Object Composition Petri Net (OCPN) model based on the logic of temporal intervals and Timed Petri Nets which was proposed to store, retrieve, and communicate between multimedia objects. This model, which is a modification of earlier F'etri net models, consists of a set of transitions (bars), a set of places (circles), and a set of directed arcs. Many later semantic models are based on Time Petri Nets. However, this model does not allow branching so that users cannot choose the scenario that they want to watch as discussed in Chapter 1. Empirical studies to compare the memory requirement of OCPN and ATN are shown in Chapter 3. The results show that OCPN needs more memory than our ATN model when document size increases.

Chang et al. (1995) and Lin et al. (1996) developed TAO (Teleaction object) and OEM (Object Exchange Manager). TAO is a multimedia object with associated hypergraph structure and knowledge structure, and AMS (Active multimedia system) is designed to manage the TAOs. OEM maintains and manages uniform representation and interacts with other system modules. TAO is a conceptual model which can be implemented as objects in an object-oriented system and each TAO has its own private knowledge in the AMS. TAOs are connected by a hypergraph. The multimedia data schema (MDS) is similar to OCPN which controls the synchronization between time-related data streams. Users need to create a hypergraph structure using four different links (annotation link, reference link, location link, and :synchronization link) before generating a multimedia data schema. If a multimedia presentation consists of a large number of media streams, then it is very difficult for users to construct a corresponding hypergraph structure. A multimedia communication schema (MCS) is obtained based on MDS for an efficient transmission sequence. Although the authors proposed an algorithm to translate MDS to MCS, the necessary limitations of the computer hardware, the storage size, and the bandwidth need to be known in advance to form a transmission vector (TV). For example, if the data receiving end does not have audio devices then this information needs to be known first so that the (audio object can be deleted in MCS. This design can let designers specify the necessary actions for different communication delays and different computer hardware limitations, but it cannot handle some unexpected situations such as accidental computer hardware failure. Also, the relationships among semantic objects in the image and video frames are not included in their model. Hence, there are limitations on the queries; for example, users cannot issue a query such as "Find the presentation that has an airplane in the video."

Al-Salqan and Chang (1996) developed a model which uses synchronization agents as "smart" distributed objects to deal with scheduling, integrating, and synchronizing distributed multimedia streams. This formal specification model, interoperable Petri nets, describes the agents' behavior and captures the temporal semantics. It can deal with both accurate and fuzzy scenarios. Since this model is also based on a Petri net, it has the same disadvantages as the previous two Petri net models. First, it does not model user interactions, and second, it becomes complicated when the network becomes large.

2.2 Time-Interval Based Models

Two temporal models for time-dependent data retrievals in a multimedia database management system have been proposed (Little and Ghafoor,1993). These two proposed models are restricted to having the property of monotonically increasing playout deadlines for represented objects and data retrieval synchronous algorithms. Since interval-based temporal relations are used, only the duration of each data stream is considered in their paper. The starting and ending frame numbers of each video clip are not included in these two models so that it is limited to allowing users to issue the queries related to the video frame numbers. Also, these two models do not show how to achieve the object layout.

Oomoto and Tanaka (1993) developed a video-object database system named OVID. A video-object data model provides interval-inclusion based inheritance and composite video-object operations. A video object is a video frame sequence (a meaningful scene) and it is an independent object itself. Each video object has its own attributes and attribute values to describe the content. Since this model only uses frame sequences to represent the interval, the starting time and ending time of each interval are not included in this model. This model is designed to help database queries and is not a good model for the multimedia presentation since it does not capture the user interaction delays in it.

Wahl and Rothermel (1994) propose an interval-based temporal model using the temporal equalities and inequalities between events. This model is a high-level abstract model which also works for the asynchronous events so that the starting time can be determined at the presentation time. Actually, it is designed for all those events occurring in the presentation. Thus, it does not have flexibility to allow user interactions at the presentation time.

2.3 Graphic Model

Buchanan and Zellweger (1993) proposed a system called Firefly. Each media object contains two or more events such as start events, end events, synchronization events, and asynchronous events. The start and end events are represented by two rectangular nodes, the synchronization events use circular nodes that are put between the start and end events, and the asynchronous events are denoted by circular nodes that are placed above the start

events. This Firefly model becomes difficult to manage when the presentation contains many media objects that are mixed with the synchronization and asynchronous events.

Yeo and Yeung (Yeo and Yeung, 1997) proposed a video browsing model.. They developed mechanisms to classify video, find story unit, and organize video units using scene transition graphs (STGs). STGs model a cluster of shots as nodes and the transitions between shots in time as edges so that the video hierarchy and the temporal relations of each video unit are preserved. Therefore, this model provides presentation and browsing capabilities to users. Users can choose a specific scenario to watch by browsing sample video frames at different granularities. However, this model does not provide query capability to let users select interesting topics by queries. Also, the spatial relation of semantic objects is not included in their model too.

2.4 Timeline Models

Blakowski and Huebel (1991) proposed a timeline model in which all events are aligned on a single time axis. They use "before," "after," or "simultaneous to" to represent the relationships between two events. Although this model provides a simple and graphical presentation, user interactions are not included in this model because they require a total specification of all temporal relationships among media objects. For example, a user needs to use a computer mouse to select the various scenarios based on the user's preference in a video game at runtime. In this case, it does not work because the start time of this scenario is known while the end time is unknown until the user makes the choice.

Hirzalla and Karmouch (1995) proposed a timeline model to expand the traditional timeline model to include temporal inequalities between events. They also developed a timeline tree to represent an interactive scenario. This enhanced timeline model models user actions as media objects. In a traditional timeline, the vertical axis of the timeline includes only media streams such as text, image, video or audio. In their model, a new type of media object called *choice* lets users interact with the multimedia presentation. The first disadvantage of their model is that it can only model media streams rather than semantic objects as the ATN subnetwork does. The second disadvantage is that it can only handle the uncertainty of the user selection time and not provide mechanisms to handle communication delays or out

of buffer when the data arrive too soon. The third disadvantage is that it models only the temporal relations of media streams without modeling the spatial layout of media streams into the timeline model.

3. USING ATNS AND MULTIMEDIA INPUT STRINGS TO MODEL MULTIMEDIA PRESENTATIONS

In section 1.1, we explained the need for semantic models for presentations, browsings, and database searchings in multimedia information systems. Sections **2.1** through **2.4** review some of the existing approaches to this problem. In this chapter, details of how to use augmented transition networks (ATNs) and multimedia input strings are discussed. Section 3.1 introduces the ATNs and discusses how to use an ATN to model a multimedia presentation and to incorporate with multimedia database searching. The input for an ATN which is modelecl by multimedia input strings is illustrated in section **3.2.** Section **3.3** compares ATNs with the OCPN models for multimedia presentations by conducting some empirical studies. Conclusions are in section 3.4.

3.1 The Augmented Transition Network

A multimedia presentation consists of media streams displaying together or separately across time. The arcs in an ATN represent the time flow from one state to another. An ATN can be represented diagrammatically by a labeled directed graph, called a *transition graph.* The ATN grammar consists of a finite set of nodes (states) connected by labeled directed arcs. An arc represents an allowable transition from the state at its tail to the state at its head, and the labeled arc represents the transition function. An input string is accepted by the grammar if there is a path of transitions which corresponds to the sequence of symbols in the string and which leads from a specified initial state to one of a set of specified final states. ATN differs from a finite state automata in that it permits recursion so that ATN is a *recursive transition network.* Each nonterminal symbol consists of a subnetwork which can be used to model the temporal and spatial information of semantic objects for images and video frames and keywords for texts. In addition, a subnetwork can represent another existing presentation. Any change in one of the subnetworks will automatically change the

presentation which includes these subnetworks. To design a multimedia presentation from scratch is a difficult process in today's authoring environment. The subnetworks in ATN allow the designers to use the existing presentation sequence in the archives, which makes ATN a powerful model for creating a new presentation. This is similar to the *class* in the object-oriented paradigm. Also, subnetworks can model keywords in a text media stream so that database queries relative to the keywords in the text can be answered.

Definition 3.1: An augmented transition network Ψ is a 6-tuple $(\mathrm{S}, I, \mathrm{T}, S_0, \mathrm{F}, \psi)$ where

1. $\mathrm{S} = \{S_0, S_1, \ldots, S_{n-1}\}$ is a finite set of states of the control. Each state represents all the events before this state have been accomplished so that it does not need to know the history of the past to continue the presentation.

2. I is a set from which input symbols are chosen. The input string consists of one or more m_i and cm_i where m_i and $cm_i \in$ C are separated by + where m_i and cm_i denote a media stream and compressed version of that media stream, respectively.

3. T is it transition table by permitting a sequence of actions and conditions to be specified on each arc. A presentation can be divided into several time durations based on different media stream combinations. Each combination occurrence of media streams is represented by an arc symbol. The media streams in an arc symbol are displayed concurrently on this time duration. Conditions and actions control the synchronization and quality of service (QoS) of a presentation. Therefore, the real-time situations such as network congestion, memory limitation, user interaction delay can be handled.

4. S_0 is the *initial state.* $S_0 \in$ S.

5. F is the set of *final states.*

6. ψ is the subnetwork of ATN when the input symbol contains image or video streams.

Definition 3.2: An augmented transition subnetwork ψ is a 3-tuple (s, O, t) where

1. $s \subset S$.

2. O is a set from which input symbols are chosen. The input string consists of one or more o_i where $o_i \in O$ are separated by $+$, where o_i is a semantic object.

3. $t \subset T$.

States are represented by circles with the state name inside. The state name is used to indicate the presentation being displayed (to the left of the slash) and which media streams have just been displayed. The state name in each state can tell us all the events that have been accomplished so far. Based on the state name, we can know how much of the presentation has been displayed. When the control passes to a state, it means all the events before this state are finished. A state node is a breaking point for two different events. For example, in a presentation sequence, if any media stream is changed then a new state node is created to distinguish these two events. In ATNs, when any media stream begins or ends, a new state is created and an arc connects this new state to the previous state. Therefore, a state node is useful to separate different media stream combinations into different time intervals. For example, in Figure 1.7, there are six state nodes and five arc:; which represent six time instants and five time durations, respectively. The arc symbol in the outgoing arc for each state will be analyzed immediately so that the process can continue and does not need to know the past history of the presentation. Two state nodes are connected by an arc. The arc labels can tell us which media streams or semantic objects are involved. Each arc represents a time interval. For example, if an arc label contains media streams then it means these media streams will be displayed at this time interval.

The arc types together with the notation need to be defined. We adopted the following notation and definition as in (Allen, 1995).

- Pusln arc: succeeds only if the named network can be successfully traversed. The state name at the head of arc will be pushed into a stack and the control will be passed to the named network.

- **Pop** arc: succeeds and signals the successful end of the network. The topmost state name will be removed from the stack and become the return point. Therefore, the process can continue from this state node.
- **Jump** arc: always succeeds. This arc is useful to pass the control to any state node.

Each media stream contains a feature set $\mathcal{F}$ which has all the control information related to the media stream. The definition and the meaning of each element are as follows :

Definition 3.3: Suppose there are n media streams appearing in the input symbols. Each media stream has a feature set together with it.

$\mathcal{F}_i$ = {tentative_starting_time, tentative-ending-time, starting_frame, ending_frame, window_position_X, window-position-Y, window-size-width, window_size_height, priority) where $i = 1 \ldots n$.

The meaning of each element is illustrated below :

- **tentative_starting_time:** the original media stream desired starting time.
- **tentative-ending-time:** the original media stream desired ending time.
- **starting_frame:** the starting video frame number.
- **ending_frame:** the ending video frame number.
- **window_position_X:** the horizontal distance from the upper left corner of the computer screen.
- **window_position-Y:** the vertical distance from the upper left corner of the computer screen.
- **window_size_width:** the window size width of the media stream.
- **window_size_height:** the window size height of the media stream.
- **priority:** the display priority if several media streams are to be displayed concurrently.

In addition to the recursive transition network, a table consisting of actions and conditions which are specified on each arc forms an augmented transition network. The advantage of this table is that only this table needs memory space. Hence, the multimedia transition network is just a visualization of the data structure which can be embedded in the programming implementation. Conditions and actions in the arcs in ATNs maintain the synchronization and quality of service (QoS) of a multimedia presentation by permitting a sequence of conditions and actions to be specified on each arc. The conditions are to specify various situations in the multimedia presentation. A condition is a Boolean combination of predicates involving the current input symbol, variable contents and the QoS. A new input symbol cannot be taken unless the condition is evaluated to true (T). More elaborate restrictions can be imposed on the conditions if needed. For example, if the communication bandwidth is not enough to transmit all the media streams on time for the presentation, then the action is to get the compressed version of media streams instead of the raw data. In this way, synchronization can be maintained because all the media streams can arrive on time. In addition, QoS can be specified in the conditions to maintain synchronization. The actions provide a facility for explicitly building the connections among the whole ATN. The variables are the same as the symbolic variables in programming languages. They can be used in later actions, perhaps on subsequent arcs. The actions can add or change the contents of the variables, go to the next state, or replace the raw media streams with the compressed ones, and etc. In an interactive multimedia presentation, users may want to see different presentation sequences from the originally specified sequence. Therefore, in our design, when a user issues a database query, the specification in the query tries to match the conditions in the arcs. If a condition is matched then the corresponding action is invoked. Different actions can generate different presentation sequences which are dlifferent from the original sequence.

When an ATN is used for language understanding, the input for the ATN is a sentence which consists of a sequence of words with linear order. In a multimedia presentation, when user interactions such as user selections and loops are allowed, then we cannot use sentences to be inputs for an ATN. In our design, each arc in an ATN is a string containing one or more media streams displayed at the same time. A media stream is represented by a letter

subscripted by some digits. This single letter represents the media stream type and digits are used to denote various media streams of the same media stream type. For example, T_1 means a text media stream with identification number one. A multimedia input string consists of one or more media streams and is used as an input for an ATN. Multimedia input strings also have the power to model the "or" conditions and the iterative conditions. Since the heart of an ATN is a finite state automata, any multimedia input string can be represented by an ATN.

An example of how to use ATN to model a multimedia presentation (Figure 1.1) was already shown and explained in Figure 1.4 and subsection 1.5.1. The details and definitions of the multimedia input string will be discussed in the following section. Figure 1.1 also will be used as an example to show how to use multimedia input string to model a multimedia presentation. Also, how to use multimedia input strings to model semantic objects will be discussed too.

3.2 Multimedia Input Strings as Inputs for ATNs

Basically, an ATN is used for the analysis of natural language sentences. Its input is a sentence composed of words. This input format is not suitable to represent a multimedia presentation since several media streams need to be displayed at the same time, to be overlappecl, to be seen repeatedly, etc.

Multimedia input strings adopt the notations from regular expressions (Kleene, 1956). Regular expressions are useful descriptors of patterns such as tokens used in a programming language. Regular expressions provide convenient ways of specifying a certain set of strings. In this study, multimedia input strings are used to represent the presentation sequences of the temporal media streams, spatio-temporal relations of semantic objects, and keyword compositions. Information can be obtained with low time complexity by analyzing these strings. A multimedia input string goes from the left to right, which can represent the time sequence of a multimedia presentation but it cannot represent concurrent appearance and spatial location of media streams and semantic objects. In order to let multimedia input strings have these two abilities, several modifications are needed. There are two levels which need to be represented by multimedia input strings. At the coarse-grained level, the main

presentation which involves media streams is modeled. At the fine-grained level, the semantic objects in image or video frames and the keywords in a text media stream are modeled at subnetworks. Each keyword in a text media stream is the arc label at subnetworks. New states and arcs are created to model each keyword. The details to model each level are discussed in the following subsections.

3.2.1 Definitions of Variables and Notations

The following variables and notation are used in the following subsections:

- P = total multimedia presentation time
- S = starting time of the multimedia presentation
- E = ending time of the multimedia presentation without user interactions
- N = number of media streams in the presentation
- m = media stream, m $\in$ {Audio, Image, Text, Video}
- S^m = starting time of the media stream m
- E^m = ending time of the media stream m
- i = subscript of the starting frame number in video media stream
- j = subscript of the ending frame number in video media stream
- $\Delta^m_{(S^m, E^m)}$ = time duration of the media stream m
- $\tau^v_{(i,j)}$ = frame duration of the video media stream

3.2.2 Using Multimedia Input Strings to Model Media Streams and Presentations

Two notations $\mathcal{L}$ and $\mathcal{D}$ are used to define multimedia input strings:

$\mathcal{L}$ = {A, I.,T, V} is the set whose members represent the media type, where A, I, T, V denote audio, image, text, and video, respectively.

$\mathcal{D}$ = {0, 1, ..., 9} is the set consisting of the set of the ten decimal digits.

Definition 3.4: Each input symbol of a multimedia input string contains one or more media streams which are enclosed by a parentheses and are displayed at the same time interval. A media stream is a string which begins with a letter in $\mathcal{L}$ subscripted by a string of digits in $\mathcal{D}$. For example, V_1 represents a video media stream and its identification number is one. The following situations can be modeled by a multimedia input string.

- **Concurrent:** The symbol "&" between two media streams indicates these two media streams are displayed concurrently. For example, $(T_1 \& V_1)$ represents T_1 and V_1 being displayed concurrently.

- **Looping:** $m^+ = \bigcup_{i=1}^{\infty} m^i$ is the multimedia input string of positive closure of m to denote m occurring one or more times. We use the "+" symbol to model loops in a multimedia presentation to let some part of the presentation be displayed more than once.

- **Optional:** In a multimedia presentation, when the network becomes congested the original specified media streams which are stored in the remote server might not be able to arrive on time. The designer can use the "*" symbol to indicate the media streams which can be dropped in the on-line presentation. For example, $(T_1 \& V_1^*)$ means T_1 and V_1 will be displayed but V_1 can be dropped if some criteria cannot be met.

- **Contiguous:** Input symbols which are concatenated together are used to represent a multimedia presentation sequence and to form a multimedia input string. Input symbols are displayed from left to right across time sequentially. ***ab*** is the multimedia input string of ***a*** concatenated with ***b*** such that ***b*** will be displayed after ***a*** is displayed. For example, $(A_1 \& T_1)(A_2 \& T_2)$ consists of two input symbols $(A_1 \& T_1)$ and $(A_2 \& T_2)$. These two input symbols are concatenated together to show that the first input symbol $(A_1 \& T_1)$ is displayed before the second input symbol $(A_2 \& T_2)$.

- **Alternative:** A multimedia input string can model user selections by separating input symbols with the "|" symbol. So, $(a|b)$ is the multimedia input string of ***a*** or ***b***. For

example, $((A_1\&T_1)|(A_2\&T_2))$ denotes either the input symbol $(A_1\&T_1)$ or the input symbol $(A_2\&T_2)$ to be displayed.

- **Ending:** The symbol "$" denotes the end of the presentation.

Using Figure 1.1 as an example, the total presentation time is defined as follows and the variables and notations are defined in section 3.3.1:

- Total Presentation time $\mathbf{P} = \boldsymbol{E} - \boldsymbol{S}$ = t6 - t1 where S and E are the starting time and the ending time of the multimedia presentation.

The time duration of the media streams is the following. $\boldsymbol{S^m}$ and $\boldsymbol{E^m}$ are the starting time and the ending time of the media stream m.

- Timo duration of $V_1 = \Delta^{V1}_{(S^{V1},E^{V1})} = E^{V1} - S^{V1}$
- Time duration of $V_2 = \Delta^{V2}_{(S^{V2},E^{V2})} = E^{V2} - S^{V2}$
- Timo duration of $T_1 = \Delta^{T1}_{(S^{T1},E^{T1})} = E^{T1} - \boldsymbol{S^{T1}}$
- Time duration of $T_2 = \Delta^{T2}_{(S^{T2},E^{T2})} = E^{T2} - S^{T2}$
- Time duration of $I_1 = \Delta^{I1}_{(S^{I1},E^{I1})} = E^{I1} - S^{I1}$
- Time duration of $A_1 = \Delta^{A1}_{(S^{A1},E^{A1})} = \boldsymbol{E^{A1}} - \boldsymbol{S^{A1}}$

The frame duration of the video media streams is shown below. $\tau^{v}_{(i,j)}$ is the frame duration of the video media stream v. i and j are the subscripts of the starting and ending frame numbers in a video media stream.

- Frame duration of the video media stream $V_1 = \tau^{v1}_{(i,j)} = j - i + 1$
- Frame duration of the video media streams $V_2 = \tau^{v2}_{(i,j)} = j - i + 1$

The multimedia input string for Figure 1.1 is:

$$(V_1\&T_1)(V_1\&T_1\&I_1\&A_1)(T_2\&I_1\&A_1)(V_2\&T_2\&I_1\&A_1)(V_2\&A_1)$$

In this input example, at time t1, input symbol $(V_1\&T_1)$ is read and contains V_1 (video stream 1) and T_1 (text 1) which start to play at the same time and continue:to play. At time t_2, I_1 (Image 1) and A_1 (Audio 1) begin and overlap with V_1 and T_1. The delay time for I_1 and A_1 to display is equal to duration d_1 and does not need to be specified in the multimedia input string explicitly since the multimedia input string is read from left to right and I_1 and A_1 will display when the input symbol $(V_1\&T_1)$ is processed which takes the same time as delay for I_1 and A_1. This process continues until all the input symbols are read.

3.2.3 Modifications of a Multimedia Presentation

In a multimedia information system, any multimedia presentation may need to be modified so that media streams can be added, replaced, or deleted. In this subsection, how subnetworks can help the modification process will be discussed. Figure 3.1(a) is a timeline for a multimedia presentation P_1 that has two media streams V_1 and T_1. V_1 and T_1 have the same starting time (t_1) and T_1 ends (t_6) earlier than V_1 (t_7). The multimedia input string is as follow:

$$(V_1\&T_1)(V_1)$$

Since V_1 and T_1 media streams start at the same time, a symbol containing V_1 and T_1 is created. After media stream T_1 reaches its ending time (t_6), V_1 continues to display so that a new symbol containing V_1 is created. Figure 3.1(c) is the corresponding ti-ansition network for Figure 3.1(a). There are two arcs and the arc labels are $V_1\&T_1$ and V_1.

3.2.3.1 Adding New Media Streams in a Multimedia Presentation

Figure 3.1(b) is a timeline example in which two media streams A_1 and I_1 are added to presentation P_1. The starting time of A_1 (t_2) is later than the starting time of T_1 (t_1) and the ending time of A_1 (t_5) is earlier than the ending time of T_1 (t_6). The starting time of I_1 (t_3) is later than the starting time of A_1 (t_2) and the ending time of I_1 (t_4) is earlier than the ending time of A_1 (t_5). multimedia input string. The resulting multimedia input string is:

$$(V_1\&T_1)(V_1\&T_1\&A_1)(V_1\&T_1\&A_1\&I_1)(V_1\&T_1\&A_1)(V_1\&T_1)(V_1) \qquad [3.1]$$

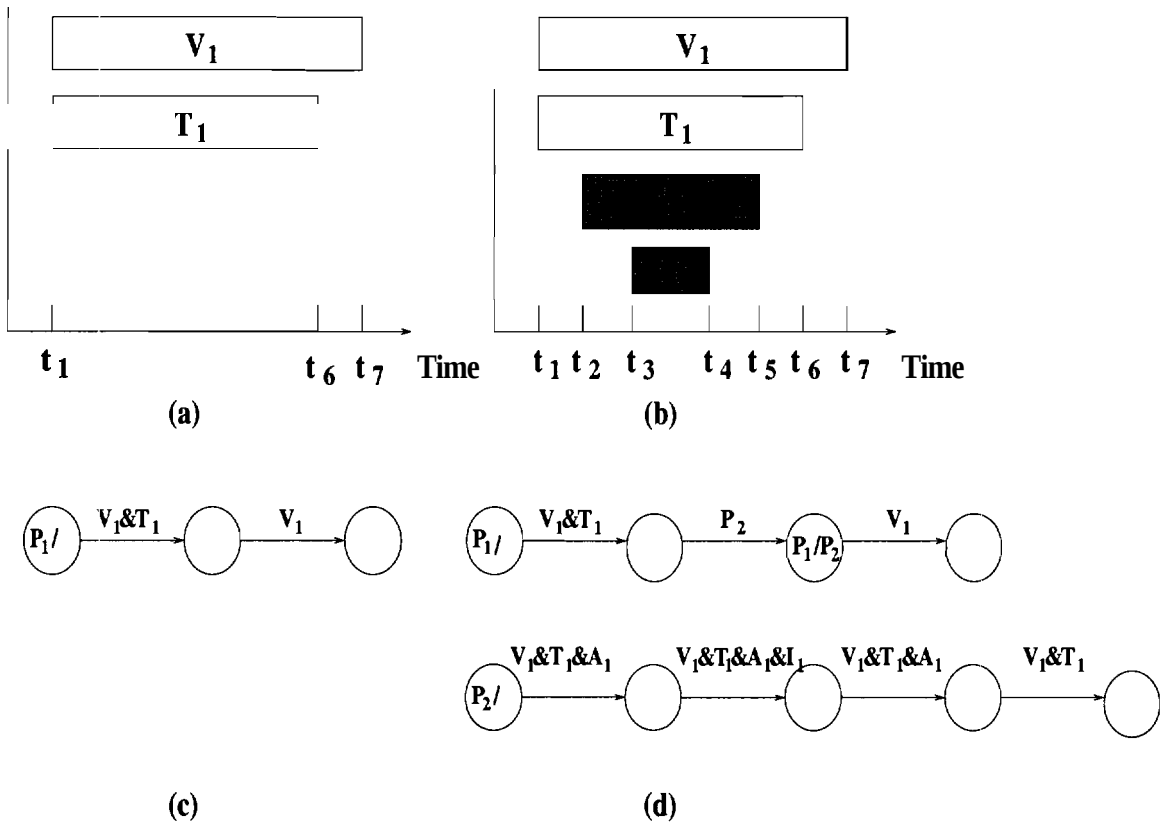


Fig. 3.1. Figure (a) is the original timeline for multimedia presentation P_1. (b) is the corresponding *transition network* for (a). Some state names are skipped in this figure and some state names are shown for explanation purposes. In (c), media streams A_1 and I_1 are added into P_1 which start at time t_2 and t_3 and end at time t_4 and t_5, respectively. (d) is the recursive transition network for (c). In (d), a new arc and a state node are created to incorporate this change. Arc label P_2 is also the starting state name of the subnetwork. There are four arcs in this subnetwork to represent four new input symbols $V_1\&T_1\&A_1$, $V_1\&T_1\&A_1\&I_1$, $V_1\&T_1\&A_1$, and $V_1\&T_1$.

In this case, four new symbols $(V_1\&T_1\&A_1)$, $(V_1\&T_1\&A_1\&I_1)$, $(V_1\&T_1\&A_1)$, and $(V_1\&T_1)$ are created to form a new multimedia input string. A subnetwork in Figure 3.1(d) can be used to incorporate this change instead of directly modifying the original *transition network* to form a *recursive transition network.* An arc with arc label P_2 is used to let the control pass to the subnetwork with the starting name P_2. There are four arcs in this subnetwork with four new symbols as their arc labels. There are two advantages of using subnetworks. First, if a presentation has several places needing to add new media streams, then a different designer can work on different subnetworks at the same time. The designer does not need to deal with the whole structure when he/she tries to change the presentation sequence. By using subnetworks, he/she can work on subnetworks first and can include subnetworks into the original *transition network* later on. Also, it is easy to include any other presentations into the original *transition network.* In order to provide the above two features, for the original *transition network,* only new arc labels with the same names as the starting state names of subnetworks are needed. This can shorten the modification processes. The same principle applies to the multimedia presentation design since we can also put part of presentations into subnetworks to allow a different designer to work on different parts independently. This advantage makes module design possible in multimedia presentation design. Second, some part of presentation may be displayed at several places; by using subnetworks then we can eliminate multiple occurrences for the same part of the presentation sequence. The reason is that the starting state name of the subnetwork can appear at several places in the *transition network.*

3.2.3.2 Deleting Media Streams in a Multimedia Presentation

After a multimedia presentation sequence has been created, designers may want to delete some media streams. Using the same example as in Figure 3.1(b), suppose we want to delete media stream T_1 so that the multimedia input string becomes:

$$(V_1)(V_1\&A_1)(V_1\&A_1\&I_1)(V_1\&A_1)(V_1) \qquad [3.2]$$

The new multimedia input string [3.2] is obtained by deleting T_1 in multimedia input string [3.1]. After deleting T_1 in [3.1], the resulting multimedia input string is:

$$(V_1)(V_1\&A_1)(V_1\&A_1\&I_1)(V_1\&A_1)(V_1)(V_1) \quad [3.3]$$

In [3.3], the last two symbols are V_1 which represents the same media stream so that these two symbols should be combined. After this combination, multimedia input string [3.2] is obtained. The same for the recursive transition network, T_1 should be deleted at the arc labels which contain it.

If we want to change Figure 3.1(b) back to Figure 3.1(a), we can just delete the arc with arc label P_2 and state node with state name P_1/P_2. The four symbols $((V_1\&T_1\&A_1)$, $(V_1\&T_1\&A_1\&I_1)$, $(V_1\&T_1\&A_1)$, and $(V_1\&T_1))$ in multimedia input string [3.1] should be deleted.

3.3 Empirical Studies comparing ATN and OCPN Models for Multimedia Presentations

In this section, a detailed comparison of ATN with Object Composition Petri Net (OCPN) (Little and Ghafoor, 1990) is shown (Chen and Kashyap, 1998(a)). OCPN is based on the logic of temporal intervals and Timed Petri-Nets. Multimedia objects are organized by the presentation sequence. The OCPN augments the conventional petri net model with time duration and resource utilization on the places in the net. Many later abstract semantic models are based on a petri-net (Chang et al., 1995; Lin et al., 1996; Al-Salqan and Chang, 1996; Thimm and Klas, 1996). All these models use nodes and arcs to connect the media streams to form a multimedia presentation. Therefore, the numbers of' nodes and arcs are essential for multimedia browsing. Since later petri-net semantic models are similar to OCPN, OCPN is chosen to be compared with ATN in this paper.

An ATN uses a multimedia input string as an input. The state nodes in an ATN do not store multimedia presentation control information; instead, the control information is stored in the multimedia transition table. A multimedia transition table is mainly for multimedia presentations. A multimedia transition table can be separated into several smaller tables based on the input symbols, so only those necessary transition tables need to be loaded into the main :memory in any real-time presentation. For the multimedia database searching, normally the query will only match the arc symbols in an ATN so that the multimedia transition table is not needed. There are several advantages to separate the multimedia transition table from an ATN. First, in the multimedia database searching, conditions to

control the multimedia presentations do not need to be checked so that it can speedup the searching time. Second, designers can change the conditions in the multimedia transition tables directly without modifying the ATN network itself.

Allen (1983) proposed thirteen temporal relations: *equal, starts, started-by, ends, ended-by, meets, met-by, contains, contained-by, overlaps, overlapped-by, before, and after.* Based on these temporal relations, two case studies are performed to compare ATN with OCPN:

- **Case study 1:** different temporal relation combinations.
- **Case study 2:** only *meets* temporal relation combination.

In case study 1, arbitrary combinations of the thirteen temporal relations of media streams are used. While in case study 2, only the *meets* temporal relation is considered, for example, a slide presentation. Under this case study, all five types of media streams. are displayed in each interval and they all have the same starting and ending times as shown in Figure 3.2. Figure 3.2 is part of a multimedia presentation in case study 2 and it contains 15 media streams with three intervals. As shown in this figure, ATN needs four nodes and three arcs and OCPN needs fifteen nodes and thirty arcs.

We wanted to count the numbers of nodes and arcs under these two case studies. The numbers of nodes and arcs plays important parts in ATN and OCPN. In OCPN, the complexity increases when the numbers of media streams and arcs increase. Media streams are assigned to state nodes which are connected by arcs. The number of arcs increases when the number of media streams increases. From the implementation point of view, more memory are required if a multimedia presentation needs more state nodes and arcs. The reason is that for each node we need to build up a structure to hold the necessary information and links need to be created to connect two state nodes. Also, the designer will have difficulty to do the :modifications and the user will have difficulty in understanding the presentation sequence under so many nodes and arcs. General observations using counting experiments to compare ATN and OCPN based on different numbers of media streams are performed in this study to show that ATN requires fewer nodes and arcs to represent the same multimedia presentation than OCPN does.

We used random number generators to generate five multimedia presentations that contain 25, 50, 100, 1000, 2000, 3000, 4000, and 5000 media streams. Four media types – text, image, audio, and video – are studied here. The combinations of media types for case study 1 and 2 are shown in Table 3.1. For example, when the media stream number is 1000 in case study 1, there are 270 text streams, 266 image streams, 234 audio streams, and 230 video streams. Each media stream has its tentative starting time and ending time so the duration is obtained:. In case study 2,

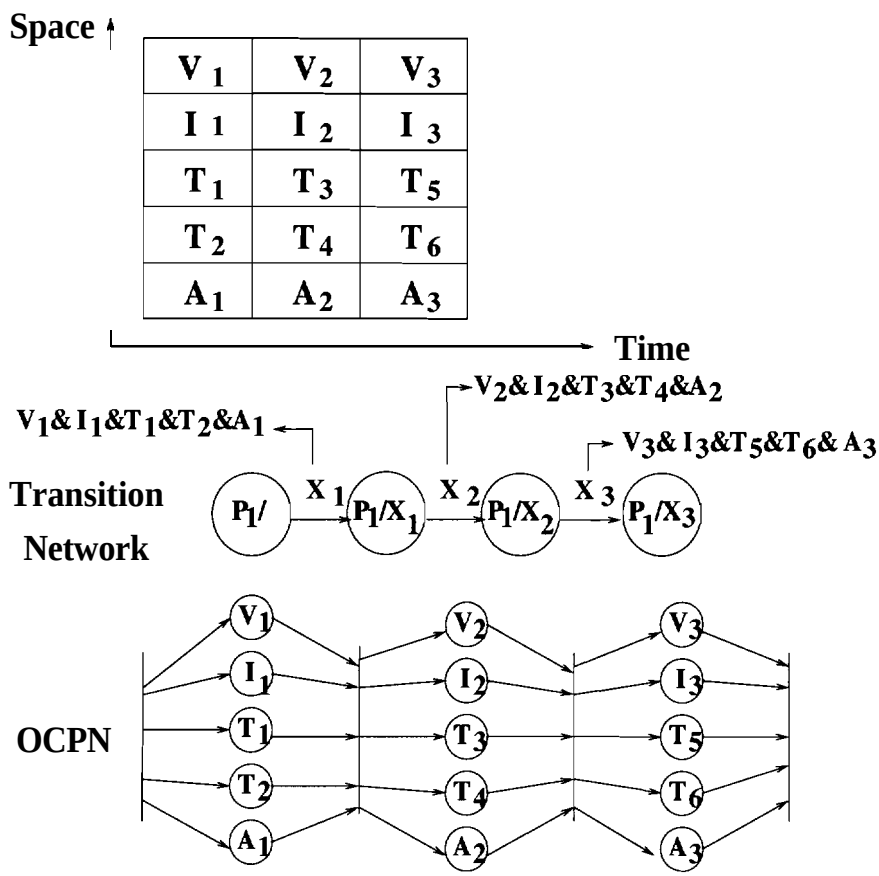


Fig. 3.2. A multimedia presentation contains 15 media streams with three intervals and each interval contains five media streams displayed at the same time for which the starting and ending times are the same.

Table 3.1 Media type combinations at different media stream numbers for **case study 1** and **case study 2**

Experiment number	Total number of media streams	Number of **Text** streams		Number of **Image** streams		Number of **Audio** streams		Number of **Video** streams	
		Case 1	Case 2	Case 1	Case 2	Case 1	Case 2	Case 1	Case 2
1	25	9	10	9	5	3	5	4	5
2	50	12	20	15	10	11	10	12	10
3	100	21	40	28	20	29	20	22	20
4	1000	270	400	266	200	234	200	230	200
5	2000	497	800	522	400	481	400	500	400
6	3000	740	1200	764	600	762	600	734	600
7	4000	987	1600	1030	800	998	800	985	800
8	5000	1249	2000	1281	1000	1252	1000	1218	1000

each duration contains one video, image, audio streams and two text streams and each media stream has the same starting and ending times in each duration. We want to compare how many nodes and arcs are needed under ATN and OCPN approaches.

3.3.1 Observation

The observation results shown in Tables **3.2** and **3.3** compare the number of nodes and number of arcs between the ATN and OCPN approaches under two case studies. From these two tables, it can be seen that ATN needs fewer nodes and arcs than OCPN in eight experiments under both case studies.

In case study 1, the number of nodes needed in ATN actually is very close to the number of media streams. The number of arcs needed in ATN is about double the number of media streams. However, since each node in OCPN has an incoming arc and an outgoing arc, the number of arcs is actually twice the number of nodes. When the media stream number increases, the difference between the number of nodes and the number of arcs increases, too. In case study **2,** under different numbers of media streams, OCPN needs about *5* times more nodes than ATN does. This tells us ATN is much better than OCPN in the case study **2** situation. The reason is that ATN creates a state node for each interval and OCPN needs to create a state node for each media stream in each interval. When comparing case studies 1 and **2** in Tables **3.2** and **3.3,** we can see that the difference between the numbers of nodes

Table 3.2 Comparison of the numbers of **nodes** between ATN and OCPN

Experiment number	Number of media streams	Number of **nodes** in ATN		Number of **nodes** in OCPN	
		Case 1	Case 2	Case 1	Case 2
1	25	32	6	38	25
2	50	55	11	69	50
3	100	104	21	132	100
4	1000	1017	201	1305	1000
5	2000	1990	401	2607	2000
6	3000	3001	601	3919	3000
7	4000	4003	801	5214	4000
8	5000	5010	1001	6536	5000

and arcs is bigger in case study 2. Also, as mentioned in Section 1.3, the numbers of nodes and arcs may grow exponentially if a lot of user selections happen in the same presentation.

From the above results, we know that when a multimedia presentation contains more media streams, ATN needs fewer nodes and arcs than OCPN does. Therefore, ATN needs less memory space and less searching time as the number of media streams increases than OCPN. An example of searching is to fast forward to a particular time point and display. All the nodes and arcs between the current time point and the target time point need to be traversed. In this situation, ATN performs better than OCPN since ATN consists of fewer nodes and arcs than OCPN.

3.4 Conclusions

In a multimedia presentation, modeling the temporal relations among media streams is very important for both designers and users. The proposed ATN can represent the temporal relations of media streams easily when associated with a multimedia input string. Counting examples are performed in this study. The results show that ATN needs fewer nodes and arcs than OCPN at different numbers of media streams. This makes ATN handle real-time multimedia presentations with less main memory space. Also, any editing of the original presentation sequence is easier because fewer nodes and arcs need to be dealt with.

Table **3.3** Comparison of the numbers of **arcs** between ATN and OCPN

Experiment number	Number of media streams	Number of **arcs** in ATN		Number of **arcs** in OCPN	
		Case 1	Case 2	Case 1	Case 2
1	25	62	10	76	50
2	50	108	20	138	100
3	100	206	40	264	200
4	1000	2032	400	2610	2000
5	2000	3978	800	5214	4000
6	3000	6000	1200	7838	6000
7	4000	8004	1600	10428	8000
8	5000	10018	2000	13072	10000

4. USING ATNS AND MULTIMEDIA INPUT STRINGS TO MODEL MULTIMEDIA DATABASE SEARCHING

The chapter is organized as follows. An introduction to multimedia database searching is given. The modeling of the spatial and temporal relations of semantic objects is presented in section 4.2. In section 4.3, the formalization of searching strategies is discussed. Also, the temporal, spatial, and spatio-temporal queries in multimedia database searchings are illustrated.. Section 4.4 concludes this chapter.

4.1 Introduction

In a multimedia information environment, users may want to watch part of a presentation by specifying some features relative to image or video content prior to a multimedia presentation, and a designer may want to include other presentations in a presentation. In order to meet these two requirements, ATNs use a pushdown mechanism that permits one to suspend the current process and go to another state in the subnetwork to analyze a query that involves temporal, spatial, or spatio-temporal relationships. Subnetworks are separated from the main ATN. Before control is passed to the subnetwork, the state name at the head of the arc is pushed into the push-down store (stack). The analysis then goes to the subnetwork whose initial state name is part of the arc label. When a final state of the subnetwork is reached, the control goes back to the state removed from the top of the push-down store.

Three situations can generate subnetworks. In the first situation, when an input symbol contains an image or a video frame, a subnetwork is generated. A new state is created for the subnetwork if there is any change in the number of semantic objects or any change in the relative position. Therefore, the temporal, spatial, or spatio-temporal relations of the semantic objects are modeled in this subnetwork. In other words, users can choose the scenarios relative to the temporal, spatial, or spatio-temporal relations of the video or image contents that they want to watch via queries. Second, if an input symbol contains

a text meclia stream, the keywords in the text media stream become the input symbols of a subnetwork. A keyword can be a word or a sentence. A new state of the subnetwork is created for each keyword. Keywords are the labels on the arcs. The input symbols of the subnetwork have the same order as the keywords appear in the text. Users can specify the criteria based on a keyword or a combination of keywords in the queries. In addition, information of other databases can be accessed by keywords via the text subnetworks. For example, if a text subnetwork contains the keyword "Purdue University Library" then the Purdue University library database is linked via a query with this keyworcl. In this design, an ATN can connect multiple existing database systems by passing control to them. After exiting the linked database system, the control is back to the ATN. Third, if an ATN wants to include another existing presentation (ATN) as a subnetwork, the initial state name of the existing presentation (ATN) is put as the arc label of the ATN. This allows any existing presentations to be embedded in the current ATN to make a new design easier. The advantage is that the other presentation structure is independent of the current presentation structure. This makes both the designer and users have a clear view of the presentation. Any change in the shared presentation is done in the shared presentation itself. There is no need to modify those presentations that use it as a subnetwork.

This chapter will focus on how to use multimedia input strings and ATN to model the temporal and spatial relations of semantic objects. In our design, each image or video frame has a subnetwork which has its own multimedia input string. Subnetworks and their multimedia input strings are created by the designer in advance for a class of applications. Users can issue multimedia database queries using high-level database query languages such as SQL. Each high-level query then translates into a multimedia input string so that it can match with the multimedia input strings for subnetworks. Therefore, database queries become the substring matching processes. A multimedia input string is a left to right model which can model the temporal relations of semantic objects. The semantic objects in the left input symbol appear earlier than those in the right input symbol in a video sequence. The spatial locations of semantic objects also need to be defined so that the queries relative to spatial locations can be answered. In our design, the temporal and spatial relations of semantic objects of a video stream in each input symbol can be modeled by a multimedia

input string. User queries can be answered by analyzing the multimedia input string (for example, the movement, the relative spatial location, the appearing sequence, etc. of semantic objects). The spatial location of each semantic object needs to be represented by a symbolic form in order to use multimedia input strings to represent it.

4.2 Modeling the Spatial and Temporal Relations of Semantic Objects

Spatial data objects often cover multi-dimensional spaces and are not well represented by point locations. It is very important for a database system to have an index mechanism to handle spatial data efficiently, as required in computer aided design, geo-data applications, and multimedia applications. R-tree (Guttman, 1984), which was proposed as a natural extension of B-trees (Bayer and McCreight, 1970; Comer, 1979), combines the nice features of both B-trees and quadtrees. An R-tree is a height-balanced tree similar to a B-tree. The spatial objects are stored in the leaf level and are not further decomposed into their pictorial primitives, i.e., into quadrants, line streams, or pixels. We call this spatial object a *"semantic object."* We adopt the minimal bounding rectangle (MBR) concept in R-ti-ees so that each semantic object is covered by a rectangle. Three types of topological relations between the MBRs can be identified (Chang et al., 1988):

(1) nonoverlapping rectangles;

(2) partly overlapping rectangles;

(3) completely overlapping rectangles.

For the second and the third alternatives, orthogonal relations proposed by Chang et al. (1988) can be used to find the relation objects. In this section, we consider only the first alternative, the nonoverlapping rectangles.

Automatic segmentation and recognition of semantic objects are outside the scope of this paper. We assume that each image or video frame has been segmented automatically or identified manually. The main focus in this paper is how to model the semantic objects so that the queries related to these semantic objects can be answered quickly.

Definition 4.1: Let O be a set of n semantic objects, $O = (o_1, o_2, \ldots, o_n)$. Associated with each o_i, $\forall i, (1 \leq i \leq n)$, is an MBR_i which is a minimal bounding rectangle

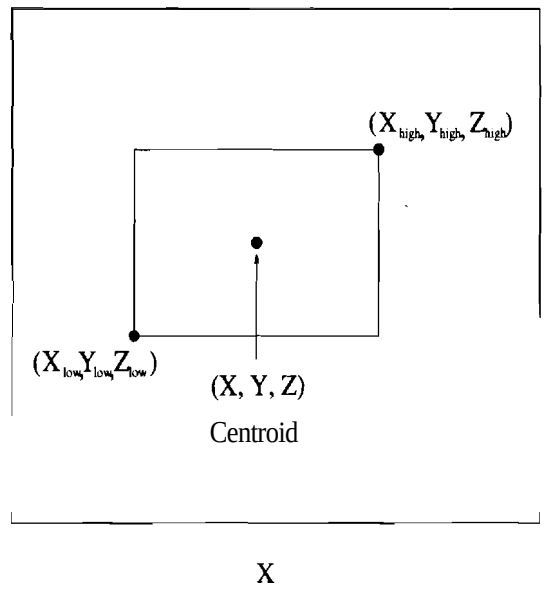


Fig. 4.1. Bounding box for semantic objects.

containing the semantic object. In a 3-D space, an entry MBR_i is a rectangle between points $(x_{low}, y_{low}, z_{low})$ and $(x_{high}, y_{high}, z_{high})$. The centroid is used as a reference point in spatial reasoning.

An example of a semantic object covered by a minimal bounding box and the centroid point is shown in Figure 4.1. Each semantic object is covered by a box between points $(x_{low}, y_{low}, z_{low})$ and $(x_{high}, y_{high}, z_{high})$. The centroid is used as a reference point for spatial reasoning. The upper-left corner is the original point for all semantic objects in a single video frame. Each minimal bounding box occupies a certain area of a video frame.

As mentioned in the previous chapter, multimedia input strings are used to represent the temporal relations among media streams. In this section, the use of multimedia input strings to represent the temporal and the spatial relations of semantic objects is described. The following definition shows the notation for the relative positions in multimedia input strings.

Definition 4.2: Each input symbol of a multimedia input string contains one or more semantic objects which are enclosed by parentheses and appear in the same image or video frame. Each semantic object has a unique name which consists of some letters. The relative positions of the semantic objects relative to the target semantic object are represented by numerical subscripts. A superscripted string of digits is used to represent different subcomponents of *relation objects* if partial or complete overlapping of MBR occurs. The "&" symbol between two semantic objects is used to denote that the two semantic objects appear in the same image or video frame.

This representation is similar to the temporal multimedia input string. One semantic object is chosen to be the target semantic object in each image or video frame. In order to distinguish the relative positions, the three dimensional spatial relations are developed (as shown in Table 4.1). In this table, twenty-seven numbers are used to distinguish the relative positions of each semantic object relative to the target semantic object. Value 1 is reserved for the target semantic object with (x_t, y_t, z_t) coordinates. Let (x,, y_s, 2,) represent the coordinate:; of any semantic object. The relative position of a semantic object with respect to the target semantic object is determined by the X-, Y-, and Z-coordinate relations. The

Table 4.1 Three dimensional relative positions for semantic objects: The first and the third columns indicate the relative position numbers while the second and the fourth columns are the relative coordinates. (x_t, y_t, z_t) and (x_s, y_s, z_s) represent the X-, Y-, and Z-coordinates of the target and any semantic object, respectively. The "$\approx$" symbol means the difference between two coordinates is within a threshold value.

Number	**Relative Coordinates**	**Number**	**Relative Coordinates**
1	$x_s \approx x_t, y_s \approx y_t, z_s \approx z_t$	15	$x_s < x_t, y_s < y_t, z_s > z_t$
2	$x_s \approx x_t, y_s \approx y_t, z_s < z_t$	16	$x_s < x_t, y_s > y_t, z_s \approx z_t$
3	$x_s \approx x_t, y_s \approx y_t, z_s > z_t$	17	$x_s < x_t, y_s > y_t, z_s < z_t$
4	$x_s \approx x_t, y_s < y_t, z_s \approx z_t$	18	$x_s < x_t, y_s > y_t, z_s > z_t$
5	$x_s \approx x_t, y_s < y_t, z_s < z_t$	19	$x_s > x_t, y_s \approx y_t, z_s \approx z_t$
6	$x_s \approx x_t, y_s < y_t, z_s > z_t$	20	$x_s > x_t, y_s \approx y_t, z_s < z_t$
7	$x_s \approx x_t, y_s > y_t, z_s \approx z_t$	21	$x_s > x_t, y_s \approx y_t, z_s > z_t$
8	$x_s \approx x_t, y_s > y_t, z_s < z_t$	22	$x_s > x_t, y_s < y_t, z_s \approx z_t$
9	$x_s \approx x_t, y_s > y_t, z_s > z_t$	23	$x_s > x_t, y_s < y_t, z_s < z_t$
10	$x_s < x_t, y_s \approx y_t, z_s \approx z_t$	24	$x_s > x_t, y_s < y_t, z_s > z_t$
11	$x_s < x_t, y_s \approx y_t, z_s < z_t$	25	$x_s > x_t, y_s > y_t, z_s \approx z_t$
12	$x_s < x_t, y_s \approx y_t, z_s > z_t$	26	$x_s > x_t, y_s > y_t, z_s < z_t$
13	$x_s < x_t, y_s < y_t, z_s \approx z_t$	27	$x_s > x_t, y_s > y_t, z_s > z_t$
14	$x_s < x_t, y_s < y_t, z_s < z_t$		

"≈" symbol means the difference between two coordinates is within a threshold value. For example, relative position number 10 means a semantic object's X-coordinate (x,) is less than the X-coordinate (x_t) of the target semantic object, while Y- and Z-coordinates are approximately the same. In other words, the semantic object is on the left of the target semantic object. More or fewer numbers may be used to divide an image or a video frame into subregions to allow more fuzzy or more precise queries as necessary. The centroid point of each semantic object is used for space reasoning so that any semantic object is mapped to a point object. Therefore, the relative position between the target semantic object and a semantic object can be derived based on these centroid points. A multimedia input string then can be formed after relative positions are obtained. Del Bimbo et al. (Bimbo et al., 1995) proposed a region-based formulation with a rectangular partitioning. Therefore, each object stands over one or more regions. Table 4.1 follows the same principle but directly captures the relative positions among objects. Relative positions are explicitly indicated by numbers to capture the spatial relations and moving history. Different input symbols in multimedia input strings represent different time durations in a video sequence. These input symbols in multimedia input strings are the arc labels for the subnetworks in ATNs. In ATNs, when an arc contains one or more images, video segments, or texts then one subnetwork with the media stream as the starting state name is created. A new arc and a new state node in a subnetwork, and a new input symbol in a multimedia input string are created when any relative position of a semantic object changes or the nurnber of semantic objects changes. The subnetwork design is similar to the VSDG model (Day et al., 1995) which uses transitions to represent the number of semantic object changes. However, in the VSDG model, a new transition is created when the number of semantic objects changes and a motion vector is taken with each node. In our design, in addition to the changes of the number of semantic objects, any relative position change among semantic objects is considered and a state node and an arc in the subnetwork and an input symbol in the multimedia input string are created for this situation. Based on our design, the temporal relations and the relative positions of semantic objects can be obtained, and the moving histories of the semantic objects in the video sequence can be kept. Therefore, substring matching processes using multimedia input strings in database queries can be conducted.

Figures 4.2 through 4.4 are three video frames with frame numbers 1, 52, and 70 which contain four semantic objects, salesman, box, file holder, and telephone. They are represented by symbols S, B, F, and T, respectively. Each semantic object is surrounded by a minimal bounding rectangle. Let salesman be the target semantic object. In Figure 4.2, the relative position numbers of the other three semantic objects with respect to the target semantic object are at 10, 15, and 24, respectively. The semantic object box moves from left to front of the target semantic object salesman in Figure 4.3, and moves back to left in Figure 4.4. The following multimedia input string can be used to represent Figures 4.2 through 4.4 as follows:

$$\text{Multimedia, input string: } \underbrace{(S_1\&B_{10}\&F_{15}\&T_{24})}_{X_1}\underbrace{(S_1\&B_3\&F_{15}\&T_{24})}_{X_2}\underbrace{(S_1\&B_{10}\&F_{15}\&T_{24})}_{X_3}, \qquad [4.1]$$

Input symbol X_1, X_2, and X_3 represents Figures 4.2, 4.3, and 4.4, respectively. S_1 in symbol X_1 means salesman is the target semantic object. B_{10} represents that the semantic object box is on the left of salesman, F_{15} means semantic object file holder is below and to the left of salesman, and so on. B_3 in symbol X_2 means the relative position of *box* changes from left to front. Semantic objects file holder and telephone do not change their positions so that these two semantic objects have the same relative position numbers in X_1, X_2, X_3. As we can see from this example, the multimedia input string can represent not only the relative positions of the semantic objects but also the motion of the semantic objects. For example, the above multimedia input string shows the semantic object box moves from left to front relative to the target semantic object salesman. Figure 4.5 is a subnetworls for multimedia input string [4.1]. We assume this subnetwork models the media stream V_1 in Figure 1.4. Therefore, the starting state name for this subnetwork is $V_1/$. As shown in Figure 4.5, there are three arcs with arc labels the same as the three input symbols in [4.1].

Fig. 4.2. Video frame 1. There are four semantic objects: *salesman, box, file holder,* and *telephone. salesman* is the target semantic object. The relative position. numbers (as defined in Table 4.1) of the other three semantic objects are in the 10, 15, and 24, respectively.

Fig. 4.3. Video frame 52. Semantic object *box* moves from the left to the front of *salesman.*

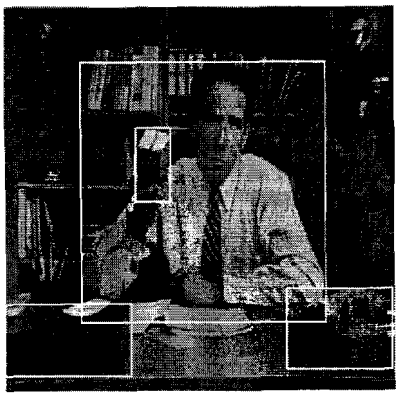


Fig. 4.4. Video frame 70. Semantic object *box* moves from the front to the left of *salesman.*

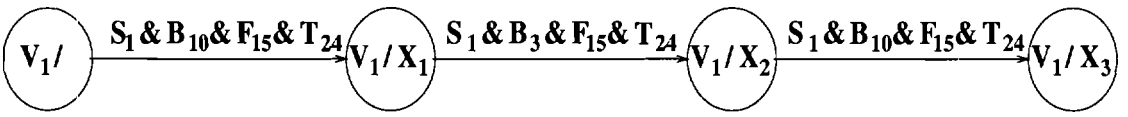


Fig. 4.5. The corresponding subnetwork for multimedia input string [4.1].

Figure 4.6 is another example showing how to use the multimedia input string to represent the relative positions. In Figure 4.6(a), there are five semantic object:; P, Q, R, S, and T surrountled by five MBRs. Let T be the target semantic object. The relative positions of the other semantic objects are at 16, 25, 13, and 22, respectively. The numbers are chosen based on the criteria defined in Table 4.1. In Figure 4.6(b), the semantic object Q disappears and the semantic object P moves to the above and to the right of T. The multimedia input string for Figure 4.6(a) and (b) is as follows:

$$\text{Multimedia input string: } \underbrace{(P_{16}\&Q_{25}\&R_{13}\&S_{22}\&T_1)}_{X_1}\underbrace{(P_{25}\&R_{13}\&S_{22}\&T_1)}_{X_2},$$

In input symbol X_1, T_1 indicates that T is the target semantic object, P_{16} means P is above and to the left of T, Q_{25} means Q is above and to the right of T, and so on. In input symbol X_2, P_{25} denotes that the relative position of P changes to the above and to the right of T. Q does not appear in input symbol X_2 indicating that Q disappears as shown in Figure 4.6(b).

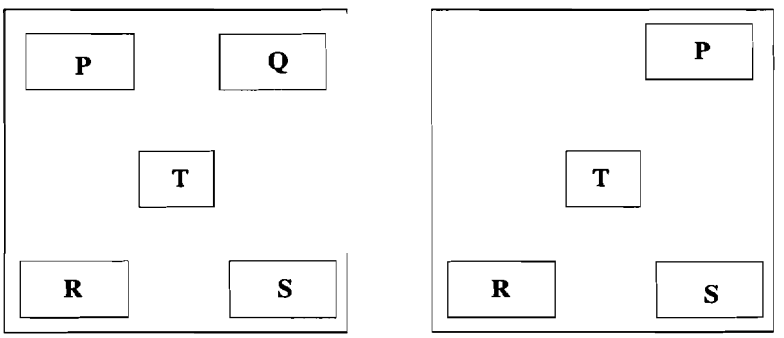


Fig. 4.6. In (a), there are five semantic objects P, Q, R, S and T. In (b), the semantic object Q disappears and the semantic object P moves to the above and to the right of T

4.3 Multimedia Database Searching

In a multimedia environment, users should be allowed to issue queries to get the information and display to them. How to combine multimedia presentations with multimedia database systems is a big challenge today. ATNs together with multimedia input strings have the ability not only to model the multimedia presentation but also to answer the multimedia database queries. In the following subsections, how to use ATNs and multimedia input strings to answer user queries relative to temporal, spatial, spatio-temporal, recursive, and unordered aspects is discussed. The most important thing is that users can specify the criteria relative to the contents of images or video frames.

4.3.1 The Formalization of Searching Strategies

Figure *4.7* is the control flow for multimedia database searching using ATN. First, a high level query is translated into a multimedia input string. For the convenience, we use $R1$ to represent this multimedia input string. $R1$ is then to check whether it related to video, image, or keyword. If not, then $R1$ matches with the multimedia input string of this ATN. Substring matching is conducted to see whether $R1$ matches part or all of the multimedia input string. If $R1$ is related to video, image, or keyword then the ATN is traversed. In ATNs, the complexity of a query depends crucially on the order in which the network is searched for a successful path. If the states are in linear order, then the traversing goes from left to right. However, when two or more arcs are leaving out of a state, the traversing order needs to be specified. Our searching strategy for this situation is to go from the topmost path to the bottommost path. When traversing an attempted arc, the remaining untried arcs leaving the state are temporarily held in a list. After this attempted arc is traversed, one alternative is removed from the front of this list and tried. The process continues until no alternatives are left in this list. As a result of this depth-first search, the information which satisfies the criteria in the queries is obtained. In order to search information related to video, image, or text, arc labels are checked to see whether they contain any of them. If the arc label does not contain video, image, or text, then the traversing continues. When an arc contains video, image, or text, the corresponding subnetwork is traversed to see whether $R1$ matches with the multimedia input string of the subnetwork. Substing matching is used

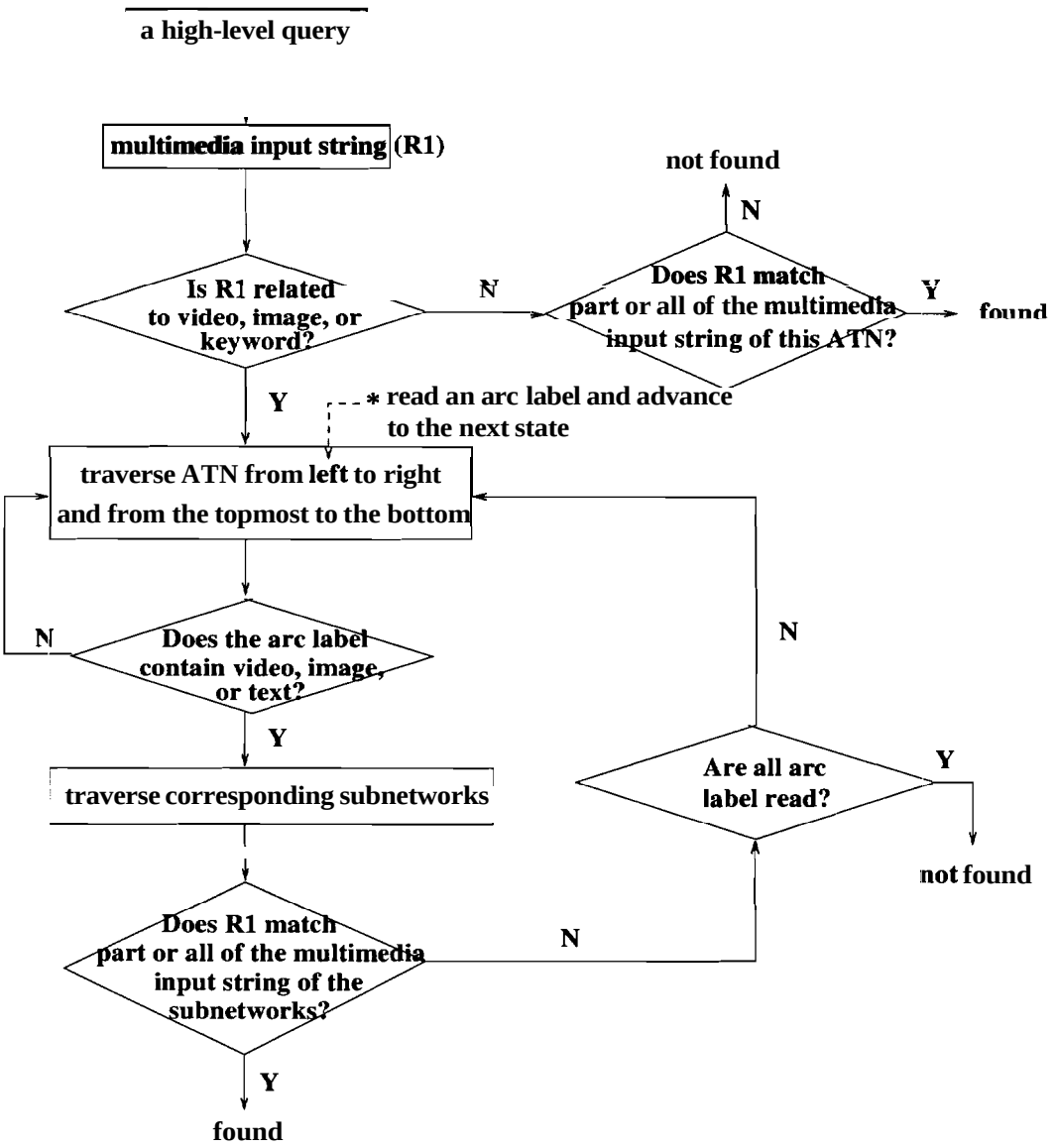


Fig. 4.7. The control flow for multimedia database searching using ATN.

to compare $R1$ with the multimedia input string for the subnetwork. The searching finishes when results are obtained or all the arc labels are compared in the ATN.

4.3.2 Examples

4.3.2.1 Temporal Database Queries

- **Query 1: Find and display the video clips beginning with a person who is alone and ending with the same person with a cat.**

In this query, we are interested in the temporal relation of two semantic objects (a *person* (R) and a *cat* (C)). This query can be translated into two multimedia input strings (R) and $(R\&C)$. Relative position number is not specified for each semantic object to indicate that the spatial location is not our concern so that it will match the semantic object at any location. For example, R matches with R_1, R_2, and so on. (R) must appear earlier than $(R\&C)$ but not necessarily immediately after (R). After being translated into a multimedia input string, this query then becomes a substring matching problem. Each subnetwork which moclels video media streams is checked to see whether this multimedia input string can be identified. These two input symbols may be found in two separate subnetworks. The conditions and actions are used to check which criteria are satisfied and to connect the required presentation parts.

4.3.2.2 A Spatial Database Query Example

In a multimedia information system, the designers can design a general purpose presentation for a class of applications so that it allows users to choose what they prefer to see using database queries. Assume there are several video media streams and V_s is one of them. The multimedia input string for the subnetwork which models V_s is the same as [4.1].

- **Query 2: Display the multimedia presentation beginning with a salesman with a box on his right.**

In this query, only the spatial locations of two semantic objects *salesman* and *box* are checked. A user can issue this query using a high-level language. This query then is translated into a multimedia input string as follows:

Multimedia input string: $(S_1 \& B_{10})$. [4.2]

The subnetworks of an ATN are traversed and the corresponding multimedia input strings are analyzed. Suppose the multimedia input string in [4.2] is to model the subnetwork for V_s. The input symbol X_1 in V_s contains semantic objects *salesman* (S), and *box* (B). Let the *salesman* be the target semantic object. The relative position of the *box* is to the left of *salesman* from a viewer's perspective. By matching [4.2] with [4.1], we know that V_s is the starting video clip of the query. When the control is passed back from the subnetwork, then the rest of the multimedia presentation begins to display.

4.3.2.3 A Spatio-Temporal Database Query Example

- **Query 3: Find the video clips beginning with a salesman holding a box on his right, moving the box from the right to his front, and ending with moving the box back to his right.**

This query involves both the temporal and spatial aspects of two semantic objects: *salesman* and *box*. This query is translated into a multimedia input string which is the same as [4.1]. Again, each of the subnetworks needs to be checked one by one. The same as in the previous query, the relative positions to be matched are based on the views that users see. The first condition in this query asks to match the relative position of the *box* to the left of the *salesman* from a viewer's perspective. When the subnetwork of V_s is traversed, S_1 and B_{10} tell us that the input symbol X_1 satisfies the first condition in which the *box* is to the left of the *salesman*. Next, the relative position of the *box* is moved from the left to the front of the *salesman*. This is satisfied by the input symbol X_2 since B_3 indicates that the *box* is to the *front* ad the *salesman*. Finally, it needs to match the relative position of the *box* to be back to the left of the *salesman*. This condition is exactly the same as the first condition and should be satisfied by the input symbol X_3. In this query, the semantic object *salesman* is the target semantic object and his position remains the same without any change.

4.3.3 Limitations

In previous subsection, we discussed how ATNs and multimedia input string can answer temporal, spatial, and spatio-temporal database queries. However, since the notations of a

multimedia input string are used to represent a high level query, our system at current stage can answer those queries which can be represented by multimedia input strings and also are related to temporal and spatial relations of semantic objects. In other words, our system can handle a subset of the multimedia input strings. Whether our system can answer the queries of all possible multimedia input strings or beyond is not discussed in this paper and needs to be further investigated.

4.4 Conclusions

From this subsection, we know that after the subnetworks and their multimedia input strings are constructed by the designer, users can issue database queries related to temporal and spatial relations of semantic objects using high-level database query languages. These queries are translated into multimedia input strings to match with those multimedia input strings of the subnetworks that model image and video media streams. Under this design, multimedia database queries related to images or video frames can be answered. The details of the multimedia input strings, the translation from high-level queries to multimedia input strings, and the matching processes are transparent to users. ATNs and multimedia input strings arc: the internal data structures and representations in DBMS. After users issue queries, the latter processes are handled by DBMS. Separating the detailed internal processes from users can reduce the burden of users so that the multimedia information system is easy to use.

5. USING ATNS AND MULTIMEDIA INPUT STRINGS TO MODEL EMBEDDED PRESENTATIONS, USER INTERACTIONS, AND LOOP'S

In this chapter, examples of using ATNs and multimedia input strings to model embedded presentations, user interactions, loops, synchronization, and QoS maintenance in multimedia presentations are discussed.

The organization of this chapter is as follows. First, we define some key terms and introduce the notations used in this chapter. Section 5.2 gives an example using the Timeline model for multimedia presentations. We then detail how the ATN and the multimedia input string are used to model the example introduced in the previous section. In section 5.4, another example illustrates that the ATN and the multimedia input strings can model user interactions and loops. The conclusion is in section 5.5.

5.1 Definitions of Variables and Notations

The following variables and notation are used in the following subsections:

- $\mathcal{P}$ = total multimedia presentation time
- S = starting time of the multimedia presentation
- $\mathcal{E}$ = ending time of the multimedia presentation without user interactions
- δ_k = kth user interaction delay time, k $\in \{1, 2, 3, \ldots, \}$
- $\mathcal{TD}$ = total user interaction delay time.

5.2 A Timeline Example to Model Multimedia Presentations

Figures 5.1(a) through 5.1(g) are the timelines for two multimedia presentations. Figures 5.1(a) and 5.1(b) are the starting timelines for presentations P_1 and P_2, respectively. In

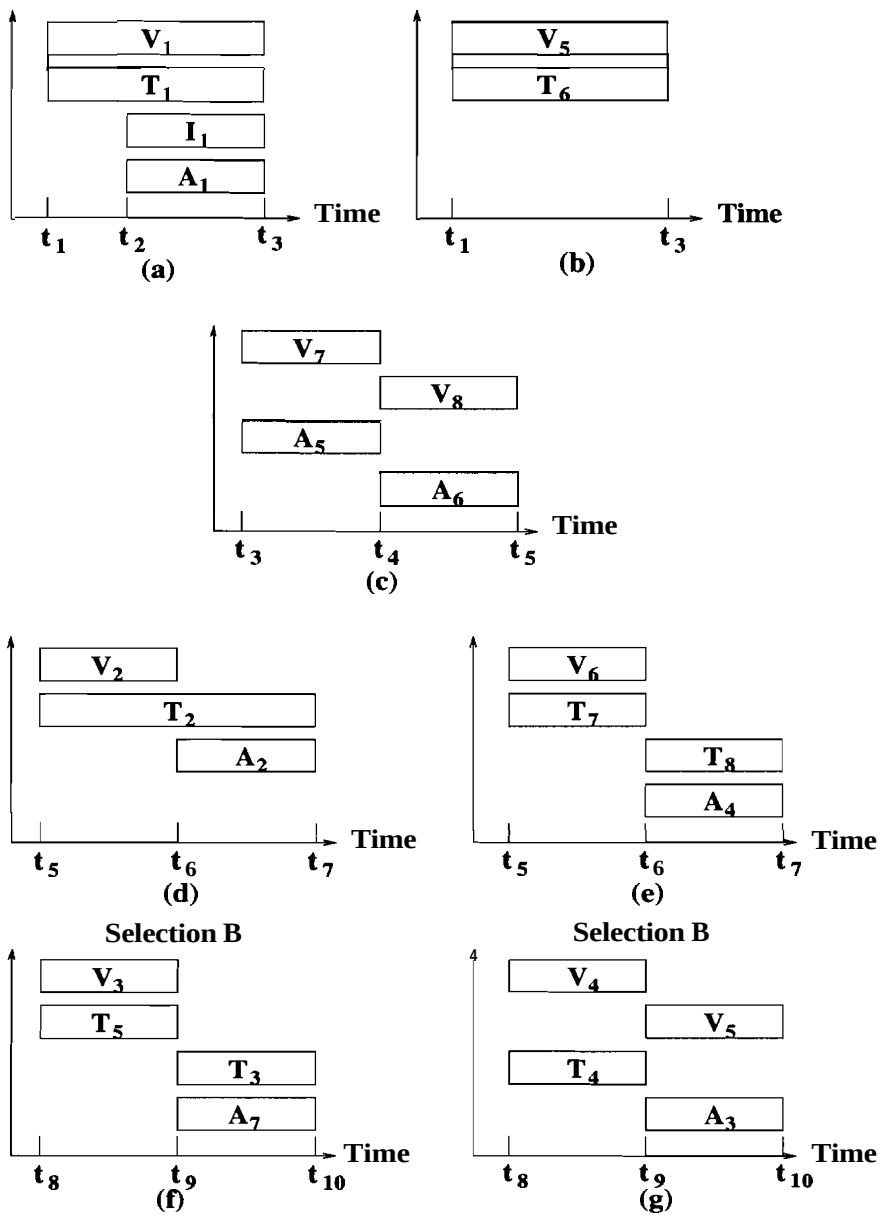


Fig. 5.1. Timelines for presentation P_1 and P_2: Figures (a), (c), (d), (e), and (g) are the presentation sequence for presentation P_1. Figures (b), (c), (e), (f), and (g) are the presentation sequence for presentation P_2. Figure (c) is an embedded presentation which is shared by both presentations. Figures (f) and (g) are two timelines for selections B_1 and B_2, respectively.

Figure 5.1(a), media streams V_1 (video stream 1) and T_1 (text 1) start to display at t_1 and media streams I_1 (image 1) and A_1 (audio stream 1) join to display at time t_2 in presentation P_1. These four media streams all end at time t_3. In presentation P_2, as shown in 5.1(b), V_5 and T_6 start at t_1 and end at t_3. At time t_3, presentations P_1 and P_2 both display V_7 and A_5 as shown in Figure 5.1(c). At time t_4, V_7 and A_5 finish and V_8 and A_6 start to display and end at time t_5. At time t_5, V_2 and T_2 begin to display in presentation P_1 as shown in Figure 5.1(d). V_2 ends at time t_6 with A_2 displayed together with T_2. T_2 and A_2 end at time t_7. Figure 5.1(e) is for presentation P_2, at time t_5, V_6 and T_7 display and end at time t_6. T_8 and A_4 then fallow V_6 and T_7 and end at time t_7. Presentations P_1 and P_2 join again at time t_7 where two choices are provided to allow users to choose based on their preference. A timeline model is inapplicable to model the alternative situation so that it is difficult to model this user interaction scenario. As shown in Figure 5.1, we cannot tell directly that Figures 5.1(f) and 5.1(g) are two timelines for different selections. Since user thinking time is unknown in advance, the starting time for media streams V_2 and T_2 and the starting time for V_4 and T_4 will not be known until the user makes a choice. Let's assume the user makes a choice at time t_8. Figures 5.1(f) and 5.1(g) are the timelines for selections B_1 and B_2. Selection B_1 has V_3 and T_2 displayed at time t_8. These two media streams end at time t_9 where T_3 and A_2 begin to display and end at time t_{10}. Selection B_2 has V_4 and T_4 displayed from time t_8 to time t_9, and V_5 and A_3 start at time t_9 and end at time t_{10}. If B_1 is chosen, the presentation stops. However, if B_2 is chosen, it allows the user to make the choice again. The timeline representation cannot reuse the same timeline in Figures 5.1(f) and 5.1(g), and therefore the same information needs to be created again. Since we do not know how many loops users go through in this part, it is impractical to use this stand alone timeline representation to model user loops.

5.3 ATNs and Multimedia Input Strings for Modeling Multimedia Presentations

When the designer designs the start and end times of media streams as shown in Figure 5.1, the multimedia input string can be constructed automatically based on the starting and ending time of media streams. In presentation P_1, the multimedia input string is:

$$\underbrace{(V_1^1\&T_1)}_{X_1}\underbrace{(V_1^2\&T_1\&I_1\&A_1)}_{X_2}\underbrace{(P_3)}_{P_3}\underbrace{(V_2\&T_2)}_{X_4}\underbrace{(T_2\&A_2)}_{X_5}\underbrace{((B_1\&B_2)}_{X_8}\underbrace{((T_2\&V_3)}_{X_9}\underbrace{(A_2\&T_3)}_{X_{10}}\ |\underbrace{(T_4\&V_4)}_{X_{11}}\underbrace{(A_3\&V_5)}_{X_{12}}))^+ \quad [5.1]$$

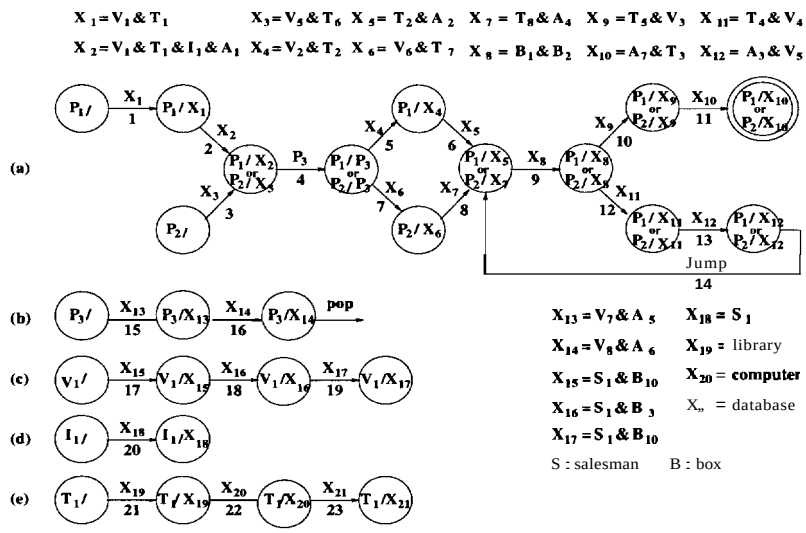
In presentation P_2, the multimedia input string is:

$$\underbrace{(V_5\&T_6)}_{X_3}\underbrace{(P_3)}_{P_3}\underbrace{(V_6\&T_7)}_{X_6}\underbrace{(T_8\&A_4)}_{X_7}\underbrace{((B_1\&B_2)}_{X_8}\underbrace{((T_2\&V_3)}_{X_9}\underbrace{(A_2\&T_3)}_{X_{10}}\ |\underbrace{(T_4\&V_4)}_{X_{11}}\underbrace{(A_3\&V_5)}_{X_{12}}))^+. \quad [5.2]$$

As mentioned earlier, a multimedia input string is used to represent the presentation sequence. In presentation P_1, the input symbol X_1 contains V_1 and T_1 which start at the same time and play concurrently. Later, I_1 and A_1 begin and overlap with V_1 and T_1. Therefore, the input symbol X_2 contains the media streams V_1, T_1, I_1, and A_1. The delay time for I_1 and A_1 to display needs not to be specified in a multimedia input string explicitly since the multimedia input string is read from left to right so that the time needed to process X_1 is the same as the delay time for I_1 and A_1. The presentation continues until the final state is reached. Each image, video, or text media stream has its own rnultimedia input string and a subnetwork is created. Figures 5.2(b) to 5.2(e) are part of the subnetworks of P_1 to model V_1, I_1, and T_1, respectively. For simplicity, the subnetworks of other image, video, and text media streams are not shown in Figure 5.2. In Figure 5.2(d), there is only one input symbol for the subnetwork modeling I_1. The input symbol is X_{18} which contains the semantic object salesman. Figure 5.2(e) is the subnetwork for T_1. The input symbols for T_1 consist of three keywords with appearing sequence library, computer, and *database*. The "|" symbol represents the alternatives for different selections. The "$" symbol denotes the end of a presentation. Figure 5.2 shows an example to use a single ATN to model two multimedia presentations which include user interactions, loops, and embedded presentations. Figure 5.2(a) is an ATN to model presentation P_1 and P_2. P_1 and P_2 start at different starting states. Table 5.1 is a trace of ATN for presentation P_1 in Figure 5.2 and is used to explain how ATN works to model embedded presentations, user interactions, and loops.

Step 1: The current state is P_1 and the arc to be followed is arc number 1 with arc label X_1. Media streams V_1 and T_1 are displayed. There is no backup state in the stack.

Step 2: The current state is P_1/X_1 which denotes X_1 has been read in presentation P_1. Arc number 2 with arc label X_2 is the arc to be followed. X_2 consists of media streams V_1, T_1, I_1, and A_1.



(f)

Arc	Symbol	Condition	Action
1	X_1	Bandwidth c Θ	Get CV_1
		Bandwidth >=Θ	Get V_1
		Current–time - Start_time(X_1) <= Duration	Display
		Current–lime - Start_time(X_1) > Duration	Next_Symbol(X_2) and Next_State
9	X_8	If choice = B,	Deluy = Current_time - Start_tirne(X_8) Next_Symbol(X_9) and Next–State
		If choice = B_2	Deluy = Current–time - Start_tirne(X_8) Next_symbol(X_{11}) and Next_State
10	X_9	Current–lie - Start_time(X_9) + Deluy <= Duratlon	Display
		Current–llme - Start_time(X_9) + Deluy >Duration	Next_symbol(X_{10}) and Next–State
11	X_{10}	Current_time - Start_time(X_{10}) + Delay <= Duration	Display
		Current–llme - Start_time(X_{10}) + Deluy >Duration	Stop
12	X_{11}	Current–llme - Start_time(X_{11}) + Deluy <= Duration	Display
		Current–lime - Start_time(X_{11}) + Deluy > Duration	Next_symbol(X_{12}) and Next_State
13	X_{12}	Current–llme - Start_time(X_{12}) + Deluy <= Duration	Display
		Current_time - Start_time(X_{12}) + Deluy > Duration	Jump and Next_symbol(X)

Fig. 5.2. Augmented Transition Network: (a) is the ATN network for two multimedia presentations which start at the states $P_1/$ and $P_2/$, respectively. (b)-(f) are part of the subnetworks of (a). (b) is an embedded presentation. (c) and (d) model the semantic objects in video media stream V_1, (e) models the semantic objects in image media stream I_1, and (f) models the keywords in text media stream T_1. In (g), CV_1 stands for the compressed version of the video media stream V_1. The "Get" procedure is to access an individual media stream. "Display" procedure is to display the media streams. "Next_Symbol(X_i)" reads the input symbol X_i. "Next_State" is a procedure to advance to the next state. "Make_choice(B_i)" is a procedure to let users make the choice which is B_i. User thinking time is kept by **delay** variable. θ is a parameter.

Step 3: In presentation P_1, after X_1 and X_2 are displayed, the ATN reaches state P_1/X_2 which has an outgoing arc (arc number 4) with an arc label P_3. The input symbol P_3 represents an existing presentation which is also a subnetwork name (as shown in Figure 5.2(b)). Since input symbol P_3 is a subnetwork name, the state name (P_1/P_3) at the head of arc 4 is put into a stack which is shown at backup states in Table 5.1. A stack follows the last-in-first-out (LIFO) policy which only allows retrieving the topmost state name first. The control passes to the subnetwork P_3 (Figure 5.2(b)) after the state name is put into stack.

The production of high-quality multimedia data such as images and video clips takes a lot of time and is expensive. It is a good way to store these data into large shared databases. Any multimedia application that uses these databases can be greatly facilitated by an underlying model that supports the development process (Schloss and Wynblatt, 1995). The main focus of this underlying model is that the data and structure can be shared, portable, and reused by the designer to create a new multimedia application. Under this design, a multimedia application (presentation) can be separated into several modules based on its conceptual structure. A finite state machine (FSM) does not have a pushdown mechanism which suspends a current process and goes to another state. Hence, it is not capable of modeling embedded presentations. However, in ATNs, the recursive control mechanisms can be used to model embedded presentations.

Step 4: The current state is $P_3/$ which is the starting state of a subnetwork as shown in Figure 5.2(b). Arc number 15 is followed and media streams V_7 and A_5 are displayed. The backup state in stack is P_1/P_3.

Step 5: The current state is P_3/X_{13} and media streams V_8 and A_6 are displayed. After these two media streams are displayed, a pop arc is met which will remove the toppest state name P_1/P_3 from the stack. Then the control will be passed back to the state P_1/P_3 in Figure 5.2(a).

Step 6: The state name P_1/P_3 in stack is popped out as the current state, which means the control is passed back to state P_1/P_3 in Figure 5.2(a). Arc 5 is followed and media streams V_2 and T_2 are displayed. The stack is empty at this step.

Step 7: The current state is P_1/X_4 and arc number 6 is followed. Media streams T_2 and A_2 are displayed.

Step 8: In user interactions, user thinking time delays need to be kept so that later presentations can be shifted. The cross-serial dependencies cannot be handled using finite state machines. However, the conditions and actions in ATNs have the ability to model user interactions. In Figure 5.2(a), after the state P_1/X_5 is met, the input symbol X_8 with two selections B_1 and B_2 is displayed. Before a choice is made, a thinking time should be kept. A **Delay** variable is used to represent the delay of the presentation. The "Start_time(X_i)" procedure gives the pre-specified starting time of X_i. The difference between the current time and the pre-specified starting time is the total display time so far. Since a delay time occurs after user interactions, the presentation sequence needs to be shifted by the delay. The process continues until the final state is reached. Figure 5.2(e) shows the detailed condition column and action column for user interactions. As illustrated in Figure 5.2(a), two choices B_1 and B_2 are provided to let users make their selections. If the choice is B_1, the input symbol X_9 is read by using the action Next_Symbol(X_9) and advance to the next state by using the action Next_State. Therefore, steps 9 to 11 in Table 5.2 will be followed. If B_2 is selectedl, steps 9 to 12 in Table 5.3 will be followed. In ATNs, an arc cannot be taken if the condition is false, even though the current input symbol satisfies the arc label. Since different selections can lead into different paths by specifying the conditions and action:; on the arc that provides selections, ATNs have the capability to handle separate presentation paths. At the same time, information about the previous states and structures can be passed along in the network to determine future transitions. Actions in ATNs provide a facility for explicitly building and naming tree structures. The names can be used in later actions, perhaps on subsequent arcs, to refer to their associated structures. The later actions can determine additions and changes to the contents of the tree structure

in terms of the current input symbol and the previous contents of the tree structure. In this example, though the previous contents of the tree structure are the same, different selections will result in reading different input symbols so that it will produce different tree structures. Hence, the recovered deep structure can be obtained after the presentation is finished.

Step 9: If B_1 is selected, arc number 10 is followed and media streams T_2 and V_3 are displayed (as shown in Table 5.2). If B_2 is selected, arc number 12 is followed and media streams T_4 and V_4 are displayed (as shown in Table 5.3).

Step 10: In Table 5.2, the current state is P_1/X_9, arc number 11 is followed, and media streams A_2 and T_3 are displayed since B_1 is selected. In Table 5.3, the current tate is P_1/X_{11}, arc number 13 is followed, and media streams A_3 and V_5 are displayed since B_2 is selected.

Step 11: When B_1 is selected, the current state is P_1/X_{10} and the presentation stops. When B_2 is selected, arc number 14 is followed and a "Jump" arc label is met. The control is passed back to state P_1/X_5.

Step 12. When B_2 is selected, the process will go back to Step 8 to let the user make the choice again. Steps 5 through 9 model a loop scenario which is represented by a "+" symbol in multimedia input strings [5.1] and [5.2]. The "Jump" action does not advance the input symbol but lets the control go to the pointing state. That means the "Jump" itself is not an input symbol in multimedia input strings. This feature is crucial for the designers who may want some part of the presentation to be seen over and over again. For example, in a computer-aided instruction (CAI) presentation, the teacher may want the students to view some part of the presentation until they become familiar with it.

Presentation P_2 is similar to P_1 except that X_3 is displayed before P_3, and X_6 and X_7 are displayed after P_3. That is, presentations 1 and 2 share arc 4 and arcs 9 through 14.

The following equations calculate the total delay and the new start tirne for any media stream displayed after a user interaction occurs.

Table 5.1 The trace of ATN for presentation P_1.

Step	Current State	Input Symbol	Arc Followed	Backup States
1	$P_1/$	X_1	1	NIL
2	P_1/X_1	X_2	2	NIL
3	P_1/X_2	P_3	4	P_1/P_3
4	$P_3/$	X_{13}	15	P_1/P_3
5	P_3/X_{13}	X_{14}	16	P_1/P_3
6	P_1/P_3	X_4	5	IYIL
7	P_1/X_4	X_5	6	IYIL
8	P_1/X_5	X_8	9	NIL

Table 5.2 Continuation of Table 5.1 if B_1 is chosen.

Step	Current State	Input Symbol	Arc Followed	Backup States
9	P_1/X_8	X_9	10	NIL
10	P_1/X_9	X_{10}	11	NIL
11	P_1/X_{10} (Finish)			

Table 5.3 Continuation of Table 5.1 if B_2 is chosen.

Step	Current State	Input Symbol	Arc Followed	Backup States
9	P_1/X_8	X_{11}	12	NIL
10	P_1/X_{11}	X_{12}	13	NIL
11	P_1/X_{12}	Jump	14	NIL
12	Go back to Step 8			

- kth user interaction delay time = δ_k = Current-time- Start_time(k); where Start_time(k) is the start time of the kth user interaction.
- New_start_time(X_i) after kth user interaction = Tentative_start_time(X_i) + δ_k; where the new start time for input symbol X_i is the sum of the original start time of X_i and the kth user interaction delay time δ_k.
- Total delay = $\mathcal{TD} = \sum_{i=1}^{m} \delta_k$ where m is the number of user selections.
- Total Presentation time $\mathcal{P} = \mathbf{I}$ - $\boldsymbol{S}$ + $\mathcal{TD}$ where $\boldsymbol{S}$ and $\mathbf{I}$ are the tentative start time and end time of the multimedia presentation.

The Start-.time is defined to be the time that the selection buttons are shown to let the user make the choice. The Current-time is defined to be the time that the user makes the choice. The difference between them is the delay time for the corresponding user interaction. After the delay is obtained, then the start time for the later presentation (New_start_time) is the sum of the tentative start time plus this delay time.

Figure 5.2(g) shows an example of how to use conditions and actions to maintain synchronization and QoS for the input symbol X_1. In presentation P_1, when the current input symbol X_1 (V_1&T_1) is read, the bandwidth condition is first checked to see whether the bandwidth is enough to transmit these two media streams. If it is not enough then the compressed version of V_1 will be transmitted instead V_1. Then the condition whether the pre-specified duration to display V_1 and T_1 is reached is checked. If it is not, the display continues. The start time is defined to be the time when the displaying of the media streams starts. The difference between the current time and the start time is the total display time so far. The last condition is met when the total display time reaches the pre-specified duration. In this case, the next input symbol X_2 is read. The same conditions are checked for X_2, too. The process continues until the final state is reached.

5.4 ATNS and Multimedia Input Strings for Modeling User interactions and Loops

Figure 5.3 shows a browsing graph of a mini-tour of a campus. The browsing graph consists of a set of nodes and arcs connecting them. There are seven sites in this browsing

graph: Purdue Mall, Computer Science, Chemical Engineering, Potter Library, Union, Electrical Engineering, and Mechanical Engineering. We use B;, i = 1 ... 7 to represent these seven site:'. For each site, a presentation consists of video, text, and audio media streams which are denoted by V_i, T_i, and A;. A directed arc denotes a one-way selection. For example, there is a directed arc pointing from the Mechanical Engineering building to the Purdue Mall. This means that after a user watches the presentation for the Mechanical Engineering building, he/she can immediately watch the Purdue Mall presentation. However, the opposite direction is inapplicable since there is no directed arc pointing from the Purdue Mall to the Mechanical Engineering building. The bi-direction arcs allow users to go back and forth between two locations such as the Purdue Mall and the Potter Library. For example, after a user watches the presentation for Purdue Mall, he/she can choose Computer Science, Chemical Engineering, Potter Library, Union, and Electrical Engineering buildings to watch. He/She can also watch the presentation for the Purdue Mall again. Figure **5.4** is the ATN for the browsing graph in Figure **5.3.** For simplicity, some state names are not shown in Figure **5.4..** Assume the browsing always starts from the Purdue Mall (B_1). There are seven networks in Figure **5.4** and each network represents a site. In a user interaction environment, users may start from any site to watch.

A detailed trace of the following browsing sequence is used to illustrate how the ATN works.

1. Purdue Mall,
2. Chemical Engineering,
3. Computer Science,
4. Computer Science.

In this browsing example, the Purdue Mall is the first to be visited, followed by the Chemical Engineering building. Then the Computer Science building is the next one to be watched. The Computer Science building is viewed one more time and then the tour stops. Table **5.4** shows the trace for this browsing example.

Step 1: The current state is in B_1/ where B_1 represents the Purdue Mall. The arc followed is arc number 1 and the input symbol is $V_1\&T_1\&A_1$. This input symbol denotes video 1, text 1, and audio 1 are displayed concurrently.

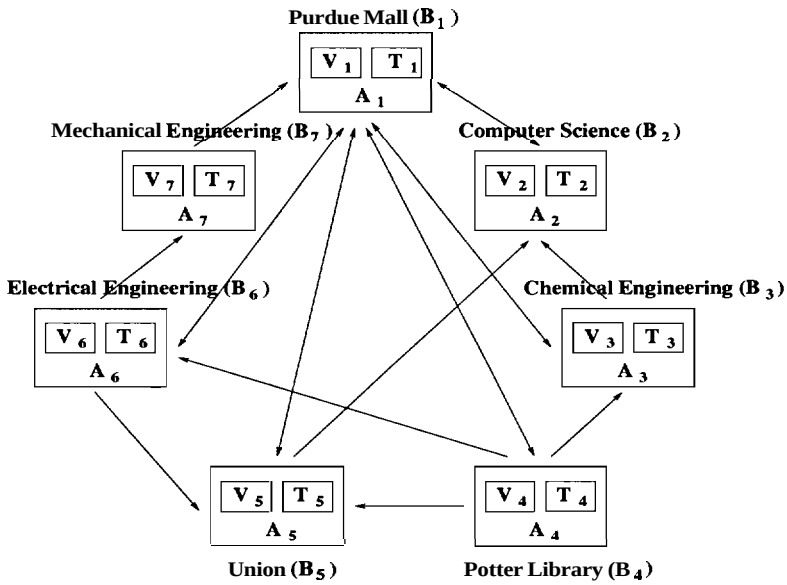


Fig. *5.3.* A browsing graph of a mini tour of the Purdue University campus: there are seven sites denoted by B_i, i = 1 ... 7 that are connected by arcs. A directed arc denotes a one-way selection and a bi-direction arc allows two-way selections.

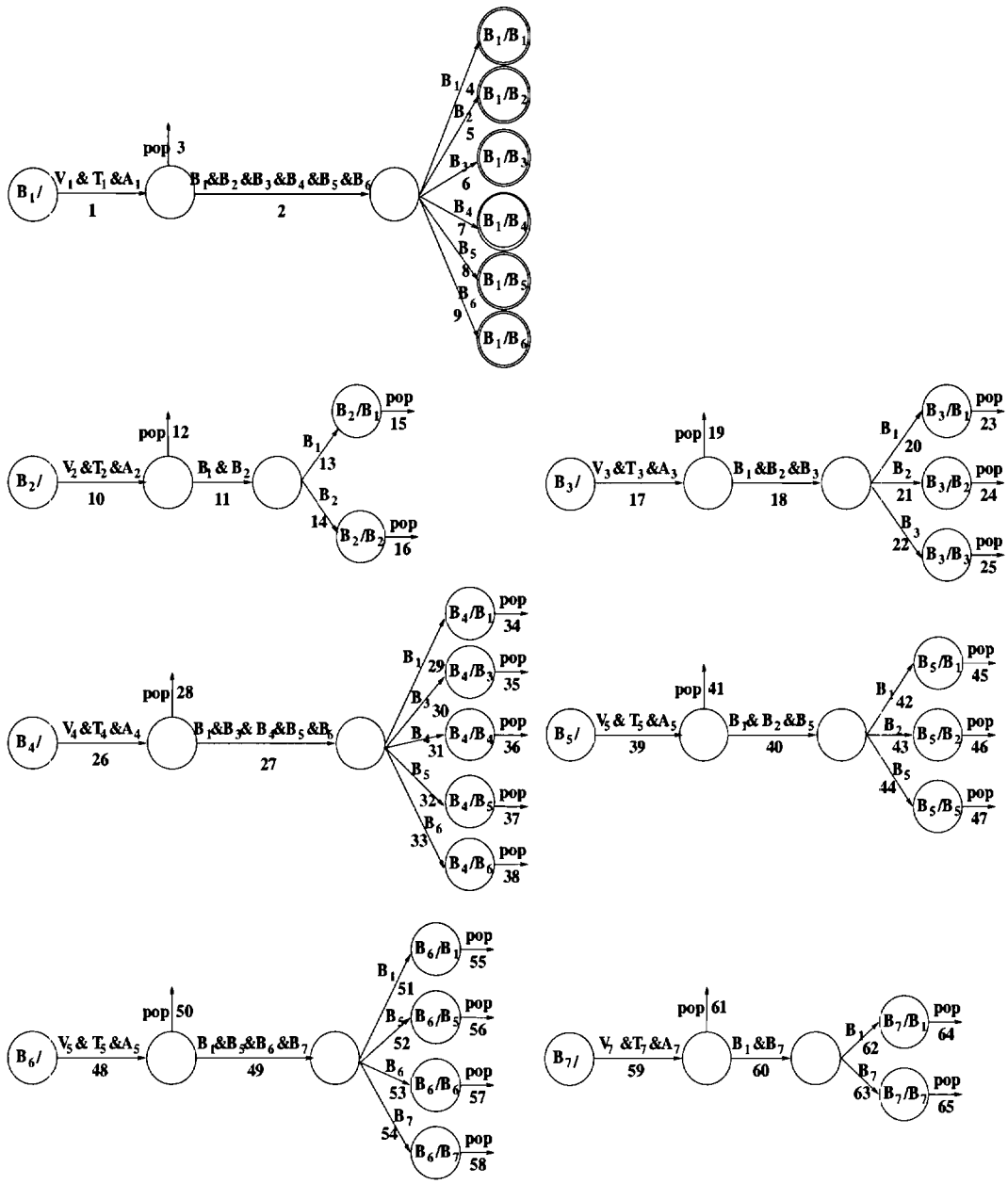


Fig. 5.4. ATN for a mini tour of the campus: Seven networks represent seven sites which users can browse. Networks $B_1/$ through $B_7/$ represent the presentations for sites Purdue Mall, Cornputer Science Building, Chemical Engineering Building, Potter Library, Union, Electrical Engineering Building and Mechanical Engineering Building respectively. Each network begins a presentation with three media streams: a video, a text, and an audio, and is followed by selections. After a user selects a site, the control will pass to the corresponding network so that the user can watch the presentation for that site continuously.

Step 2: Arc number 2 is followed and the input symbol $B_1\&B_2\&B_3\&B_4\&B_5\&B_6$ is read so that users can choose a site from site 1 through 6 to watch.

Step 3: Based on the browsing sequence as specified above, the Chemical Engineering building (B_3) is chosen so that arc number 6 is followed and input symbol B_3 is read. Since B_3 is a subnetwork name, the state name pointed by arc number 6 (B_1/B_3) is pushed into a stack. A stack follows the last-in-first-out (LIFO) policy which only allows retrieving the topmost state name first.

Step 4: The control is passed to the subnetwork with starting state name $B_3/$. Arc number 17 is followed and the input symbol $V_3\&T_3\&A_3$ is read so that video 3, text 3, and audio **3** are displayed.

Step 5: Arc number 18 is followed and the input symbol $B_1\&B_2\&B_3$ is read so that users can choose from site 1 through **3** to watch.

Step 6: As specified above, the Computer Science building (B_2) is the next site to watch so that arc number 21 is followed and the input symbol B_2 is read. Since B_2 is a subnetwork name, the state name pointed by this arc is pushed into the stack. Therefore, there are two state names in this stack: B_3/B_2 and B_1/B_3.

Step 7: The control passes to the subnetwork with starting state name $B_2/$. Arc number 10 is followed and the input symbol is $V_2\&T_2\&A_2$ so that video 2, text 2, and audio 2 are displayed.

Step 8: Arc number 11 is followed and the input symbol $B_1\&B_2$ is read. Users can choose either site 1 or 2.

Step 9: User interactions allow users to interact with multimedia information systems. Users may want to watch some topic recursively or have user loops so that they have the opportunity to select different contents after viewing a previous selection. ATNs allow recursion, that is, a network might have an arc labeled with its own name. This feature allows ATNs to have the ability to model user loops and recursion easily. As specified above, the Computer Science building (B_2) is watched again.. The state name

pointed by arc number **14** (B_2/B_2) is pushed into the stack so that there are three state names stored in this stack.

Step 10: The control passes back to the same subnetwork and video 2, text 2, and audio 2 are displayed concurrently again.

Step 11: After Step **10,** as specified above, the presentation stops so that arc number 12 is followed with a pop arc label. The topmost state name in the stack (B_2/B_2) is popped out so that the control passes to the state node with state name B_2/B_2. The stack has two state names now.

Step 12: Arc number **16** is followed with a pop arc label. Therefore the topmost state name B_3/B_2 is popped out and the control is passed to it.

Step 13: The current state name is B_3/B_2 and arc number **24** is followed with a pop arc label. The only state name in the stack is popped out and the control is passed to it.

Step 14: The current state is B_1/B_3 which is a final state (no outgoing arc) so that the browsing stops.

The ATN and its subnetworks in Figure **5.4** depict the structural hierarchy of the browsing graph in Figure **5.3.** Timeline models have difficulties in modeling user interactions and user loops because alternatives and loops are inapplicable in timeline models. On the other hand, user interactions and user loops are modeled using ATN in this example. Under ATNs, user interactions are represented by using more than one outgoing arc with different arc labels for a state node. User loops are modeled by using recursions with arcs labeled by network names. By using the recursion, one can avoid many arcs which point back to the previous state nodes. This makes the whole network structure become less complicated. Moreover, the browsing sequences in Figure **5.3** are preserved by traversing the ATN in Figure **5.4.**

5.5 Conclusions

In this chapter, we describe an ATN based model together with multimedia input strings for multimedia applications. Therefore, ATNs provide two major capabilities: *presentation*

Table 5.4 The trace of ATN for the specified browsing sequence.

Step	Current State	Input Symbol	Arc Followed	Backup States
1	$B_1/$	$V_1\&T_1\&A_1$	1	NIL
2	$B_1/V_1\&T_1\&A_1$	$B_1\&B_2\&B_3\&B_4\&B_5\&B_6$	2	NIL
3	$B_1/B_1\&B_2\&B_3\&B_4\&B_5\&B_6$	B_3	6	B_1/B_3
4	$B_3/$	$V_3\&T_3\&A_3$	17	B_1/B_3
5	$B_3/V_3\&T_3\&A_3$	$B_1\&B_2\&B_3$	18	B_1/B_3
6	$B_3/B_1\&B_2\&B_3$	B_2	21	B_3/B_2 B_1/B_3
7	$B_2/$	$V_2\&T_2\&A_2$	10	B_3/B_2 B_1/B_3
8	$B_2/V_2\&T_2\&A_2$	$B_1\&B_2$	11	B_3/B_2 B_1/B_3
9	$B_2/B_1\&B_2$	B_2	14	B_2/B_2 B_3/B_2 B_1/B_3
10	$B_2/$	$V_2\&T_2\&A_2$	10	B_2/B_2 B_3/B_2 B_1/B_3
11	$B_2/V_2\&T_2\&A_2$	pop	12	B_3/B_2 B_1/B_3
12	B_2/B_2	pop	16	B_1/B_3
13	B_3/B_2	pop	24	*NIL*
14	B_1/B_3 (Finish)			

and queryzng. ATNs are left to right models that are used to model a presentation sequence from the beginning to the end. Each arc label is a string which consists of those media streams to be displayed. A subnetwork and the corresponding multimedia input string are created for. an image, a video, or a text media stream to facilitate querying capabilities in ATNs. Subnetworks are used to model the temporal and spatial information of semantic objects for image and video sequences. The keyword sequences in text media streams can also be included in the subnetworks. Querying capabilities allow users to retrieve information related to media streams or semantic objects in a specific presentation directly. Since a great deal of research has already shown how to use ATNs to model a sentence, we do not discuss how to use ATNs to model audio media streams in this chapter. Embedded presentations allow the reusing of existing presentation sequences. This emphasizes the modularity and reuses existing media streams and presentation structures. Under this design, the storage intensive multimedia data can be stored into large shared databases, a feature which greatly reduces the design complexity and makes the design easier. This can be modeled by using subnetworks too, via putting a presentation name as an arc label such as "P_3" in Figure 5.2. Further investigation on how to solve the heterogeneity to allow searching via the embedded presentations can be conducted. User interactions are included in ATNs since ATNs provide branching for the alternative choices. User interaction features allow two-way communication between users and multimedia information systems. The user thinking times in the user's decision processes can be handled by conditions and actions in ATNs. By using a "variable" to keep track of the time duration for the decisions, the latter presentations can be shifted by this time duration since this "variable" can be passed to the later conditions to decide the adjusted starting time. Moreover, ATNs allow loops in a presentation. Loops can be used to let some part of a presentation be watched more than once.

6. VIDEO BROWSING USING ATN AND MULTIMEDIA INPUT STRINGS

After the introduction, Section 6.2 discusses how to use ATNs and multimedia input strings to model video browsing. In section 6.3, how to use recursive calls to model user loops is discussed. Key frames selection algorithm is in section 6.4. In section 6.5, how to model video unit sharing is presented. This chapter concludes with a brief summary in section 6.6.

6.1 Introduction

Unlike traditional database systems which have text or numerical data, a multimedia database or information system may contain different media such as text, image, audio, and video. Video is popular in many applications such as education and training, video conferencing, video on demand, news service, and so on. Traditionally, when users want to search certain contents in videos, they need to fast forward or rewind to get a quick overview of interest on the video tape. This is a sequential process and users do not have a chance to choose or jump to a specific topic directly. How to organize video data and provide the visual content in compact forms becomes important in multimedia applications (Yeo et al., 1997). Therefore, users can browse a video sequence directly based on their interests so that they can get the necessary information quicker and the amount of data transmission can be reduced. Also, users should have the opportunity to retrieve video materials using database queries. Since video data contains rich semantic information, database queries should allow users to get high level content such as *scenes* or *shots* and low level content according to the temporal and spatial relations of semantic objects. A semantic object is an object appearing in a video frame such as a "car." Also, a semantic model should have the ability to model visual contents at different granularities so that users can fast browse large video collections.

Many video browsing models are proposed to allow users to visualize video content based on user interactions (Arman et al., 1994; Flickner et al., 1995; Mills et al., 1992; Oomoto et al., 1993; Smoliar et al., 1994; Yeo et al. 1997). These models choose representative images using regular time intervals, one image in each shot, all frames with focus key frame at specific place, and so on. Choosing key frames based on regular time intervals may miss some important segments and segments may have multiple key frames with similar contents. One image in each shot also may not capture the temporal and spatial relations of semantic objects. Showing all key frames may confuse users when too many key frames are displayed at the same time.

In addition to using ATNs to model multimedia presentations and multimedia database searching, how to use ATNs and multimedia input strings as video browsing models is discussed in this chapter. Also, key frame selection based on the temporal and spatial relations of semantic objects in each shot will be discussed. The details on how to use the recursive call property in ATNs to model user loops are also presented.

6.2 Video Browsing Using ATNs

In interactive multimedia information systems, users should have the flexibility to browse and decide on various scenarios they want to see. This means that two-way communications should be captured by the conceptual model. Digital video has gained increasing popularity in many multimedia applications. Instead of sequential access to the video content, structuring and modeling video data so that users can quickly and easily browse and retrieve interesting materials becomes an important issue in designing multimedia information systems.

Browsing provides users the opportunity to view information rapidly since they can choose the content relevant to their needs. It is similar to the table of contents and the index of a book so that readers have a general idea by simply looking at them. The advantage is that users can quickly locate the interesting topic and avoid the sequential and time-consuming process. In a digital video library, in order to provide this capability, a semantic model should allow users to navigate a video stream based on shots, scenes, or clips. ATN can be used to model the spatio-temporal relations of multimedia presentations and multimedia database

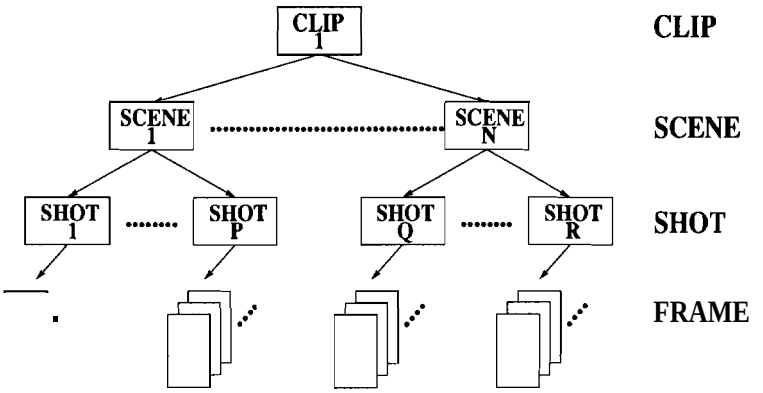


Fig. 6.1. A hierarchy of video media stream

systems. ATN allows users to view part of a presentation by issuing database queries. In this chapter, we further design a mechanism by using ATN to model video browsing so that users can :navigate the video contents. In this manner, querying and browsing capabilities can be provided by using ATNs.

6.2.1 Hierarchy for a Video Clip

As mentioned in (Yeo and Yeung, 1997), a video clip can be divided into scenes. A scene is a common event or locale which contains a sequential collection of shots. A shot is a basic unit of video production which captures between a record and a stop camera operation. Figure 6.1 is a hierarchy for a video clip. At the topmost level is the video clip. A clip contains several scenes at the second level. Each scene contains several shots. Each shot contains some contiguous frames which are at the lowest level in the video hierarchy. Since a video clip may contain many video frames, it is not good for database retrieving and browsing. How to model a video clip, based on different granularities, to accommodate browsing, searching and retrieval at different levels is an important issue in multimedia database and information systems. A video hierarchy can be defined by the following three properties:

1. v = $\{s_1, s_2, \ldots, s_N\}$, s_i denotes the ith scene and N is the number of scenes in this video clip. Let $B(s_1)$ and $E(s_1)$ be the starting and ending times of scene s_1, respectively. The temporal relation $B(s_1) < E(s_1) < B(s_2) < E(s_2) < \ldots$ is preserved.

2. $s_i = \{t_,, \ldots, t_b\}$, t_j is the jth shot in scene s_i and a and b are the shot iindices. Let $B(t_a)$ and $E(t_a)$ be the starting and ending times of shot t_a where $B(t_a) < E(t_a) < \ldots < B(t_b) < E(t_b)$.

3. $t_j = \{f_c, \ldots, f_d\}$, f_c and f_d are the starting and ending frames in shot t_j and c and d are frame indices.

In property 1, v represents a video clip and contains one or more *scenes* denoted by s_1, s_2, and so on. *Scenes* follow the temporal order. For example, the ending time of s_1 is earlier than the starting time of s_2.

As shown in property 2, each *scene* contains some *shots* such as t_a to t_b. *Shots* also follow a temporal order and there is no time overlap among shots so $B(t_a) < E(t_a) < \ldots < B(t_b) < E(t_b)$.

A *shot* contains some key frames to represent the visual contents and changes in each shot. In property **3,** f_c and f_d represent key frames c and d. The details of how to choose key frames based on temporal and spatial relations of semantic objects in each shot will be discussed in section 4.

To segrnent a video into different granularities is an active research area which is outside the scope of this study. We assume that each video clip has been segmented automatically or identified manually. The main focus in this research is to use ATNs to model the video data so that the browsing or queries related to these video units can be answered quickly. Also, how to use ATNs to share the common video units will be explored too.

6.2.2 Using ATNs to Model Video Hierarchy

An ATN can build up the hierarchy property by using its subnetworks. Figure 6.2 is an example of how to use an ATN and its subnetworks to represent a video hierarchy. An ATN and its subnetwork are capable of segmenting a video clip into different granularities and still preserve the temporal relations of different units.

Table 6.1 is a trace of ATN for presentation P_1 in Figure 6.2. The definitions and notations for different arc types in Section 3.1 are used in the following explanation. This table is used to explain how ATN works for video browsing as follows:

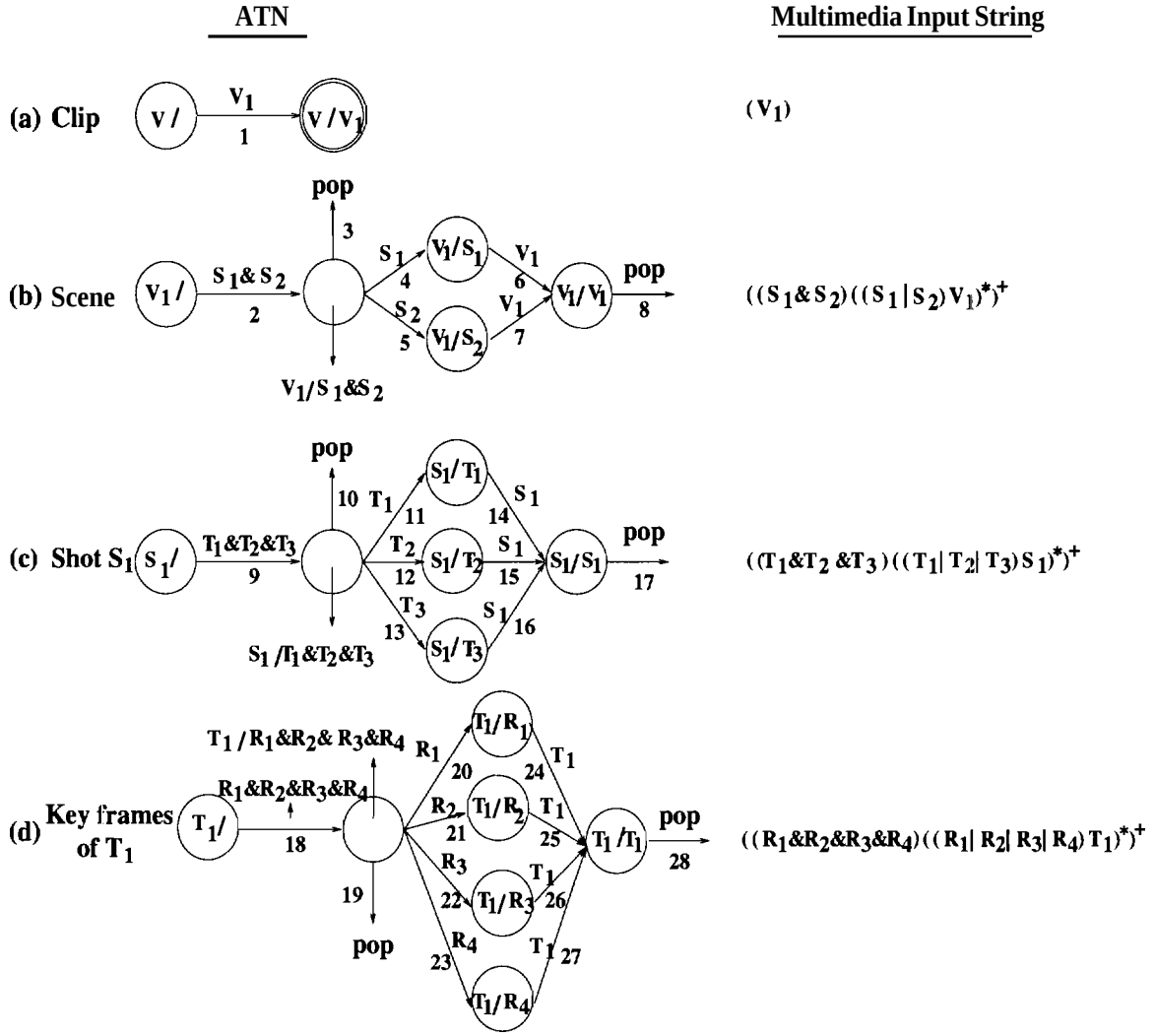


Fig. 6.2. Augmented Transition Network for video browsing: (a) is the ATN network for a video clip which starts at the state V/. (b)-(d) are part of the subnetworks of (a). (b) is to model *scenes* in video clip V_1. (c) is to model *shots* in scene S_1. Key frames for shot T_1 is in (d).

Table 6.1 The trace of ATN for the browsing sequence in Figure 6.2

Step	Current State	Input Symbol	Arc Followed	Backup States
1	$V/$	V_1	1	V/V_1
2	$V_1/$	$S_1\&S_2$	2	V/V_1
3	$V_1/S_1\&S_2$	S_1	4	V_1/S_1
				V/V_1
4	$S_1/$	$T_1\&T_2\&T3$	9	V_1/S_1
				V/V_1
5	$S_1/T_1\&T_2\&T3$	T_1	12	S_1/T_1
				V_1/S_1
				V/V_1
6	$T_1/$	$R_1\&R_2\&R3\&R_4$	18	S_1/T_1
				V_1/S_1
				V/V_1
7	$T_1/R_1\&R_2\&R3\&R_4$	R_1	20	S_1/T_1
				V_1/S_1
				V/V_1
8	T_1/R_1	T_1	24	T_1/T_1
				S_1/T_1
				V_1/S_1
				V/V_1

Continued in Table 6.2

Table 6.2 Continuation of Table 6.1

Step	Current State	Input Symbol	Arc Followed	Backup States
9	$T_1/$	$R_1\&R_2\&R_3\&R_4$	18	S_1/T_2
				V_1/S_1
				V/V_1
10	$T_1/R_1\&R_2\&R_3\&R_4$	None	19	V_1/S_1
				V/V_1
11	S_1/T_2	S_1	15	S_1/S_1
				V_1/S_1
				V/V_1
12	$S_1/$	$T_1\&T_2\&T_3$	9	S_1/S_1
				V_1/S_1
				V/V_1
13	$S_1/T_1\&T_2\&T_3$	None	10	V_1/S_1
				V/V_1
14	S_1/S_1	None	17	V/V_1
15	V_1/S_1	V_1	6	V_1/V_1
				V/V_1
16	$V_1/$	$S_1\&S_2$	2	V_1/V_1
				V/V_1
17	$V_1/S_1\&S_2$	None	3	V/V_1
18	V_1/V_1	None	8	NIL
19	Finish			

Step 1: The current state is V and the arc to be followed is arc number 1 with arc label V_1. The input symbol V_1 is a subnetwork name (as shown in Figure 6.2(b)). Since input symbol V_1 (video clip) is a subnetwork name, the state name (V/V_1) at the head of arc 1 is put into a stack which is shown at backup states in Table 6.1. The control passes to the subnetwork V_1 (Figure 6.2(b)) after the state name is put into the stack.

Step 2: The current state is $V_1/$ which is the starting state of a subnetwork as shown in Figure 6.2(b). Arc number 2 is followed and the arc label is $S_1\&S_2$. Arc label $S_1\&S_2$ means a video clip V_1 consists of two scenes to let users choose and they are S_1 and S_2. Assuming the user chooses S_1, arc number 4 is followed and the arc label (input symbol) is S_1. Since S_1 is also a subnetwork name, the state name V_1/S_1 at the head of this arc is pushed into the stack so that this state name is on top of the state name V/V_1. Therefore, there are two state names in the stack at this stage. The control passes to the subnetwork in Figure 6.2(c).

Step 3: The current state is $S_1/$. Arc number 9 with arc label $T_1\&T_2\&T_3$ is followed. This arc label denotes that scene S_1 consists of three shots: T_1, T_2, and T_3.

In Figure 6.2(a), the arc label V_1 is the starting state name of its subnetwork in Figure 6.2(b). When the input symbol V_1 is read, the name of the state at the head of the arc (V/V_1) is pushed into the top of a push-down store. The control is then passed to the state named on the arc which is the subnetwork in Figure 6.2(b).

In Figure 6.2(b), when the input symbol X_1 $(S_1\&S_2)$ is read, two frame:; which represent two video scenes S_1 and S_2 are both displayed for the selections. In the original video sequence, S_1 appears earlier than S_2 since it has a smaller number. The "&" symbol in multimedia input strings is used to denote the concurrent display of S_1 and S_2. ATNs are capable of modeling user interactions where different selections will go to different states so that users have the opportunity to directly jump to the specific video unit that they want to see. In our design, vertical bars "|" in multimedia input strings and more than one outgoing arc in each state at ATNs are used to model the "or" condition so that user interactions are allowed. Assume S_1 is selected, the input symbol S_1 is read. Control is passed to the subnetwork in Figure 6.2(c) with starting state name $S_1/$. The "*" symbol indicates the

selection is optional for the users since it may not be activated if users want to stop the browsing. The subnetwork for S_2 is omitted for the simplicity.

In Figure 6.2(c), when the input symbol $T_1\&T_2\&T_3$ is read, three frames T_1, T_2, and T_3 which represent three shots of scene S_1 are displayed for the selection.. If the shot T_1 is selected.,the control will be passed to the subnetwork in Figure 6.2(d) based on the arc symbol $T_1/$. The same as in Figure 6.2(b), the temporal flow is maintained.

6.3 User Loops

User interactions allow users to interact with multimedia information systems. Users may want to have opportunities to select different contents or to watch the same content after viewing the current program. ATNs allow partial recursion, that is, a network might have an arc labeled with its own name. This feature makes ATNs have the ability to model user loops and recursion easily. Using the same example as shown in Figure 6.2(c), when control reaches arc 16, input symbol S_1 in multimedia input string is read. The name of the state at the head of the arc (S_1/S_1) is *pushed* on the top of a push-down store and the control is passed to the starting state of the same network. This network will be traversed again to form a recursion. The "+" symbol in a multimedia input string represents this recursion. Therefore, this whole multimedia input string will be read again from the beginning. After input symbol X_2 $(T_1\&T_2\&T_3)$ is read, users can choose T_1, T_2, T_3, or stop the loop. If users decide to stop the loop then the latter input symbol will not be read. This situation is represented by the "*" symbol in multimedia input strings. The *pop* action at state S_1/X_2 then will be invoked and terminate the process at the current network. Control is returned to the state removed from the top of the push-store which is S_1/S_1. Since there is only one outgoing arc which is a *pop* action, control is again returned to the state removed from the top of the push-store. The same process continues until it reaches the final state V/V_1. The main restriction to using recursive calls is that it must start from the beginning of the same network.

Conditions and actions in ATNs can control the synchronization and quality of service of the multimedia presentation and browsing. Users can either display video contents based on an individual *scene* or can further browse at *shot* and *frame* levels while continuing to traverse

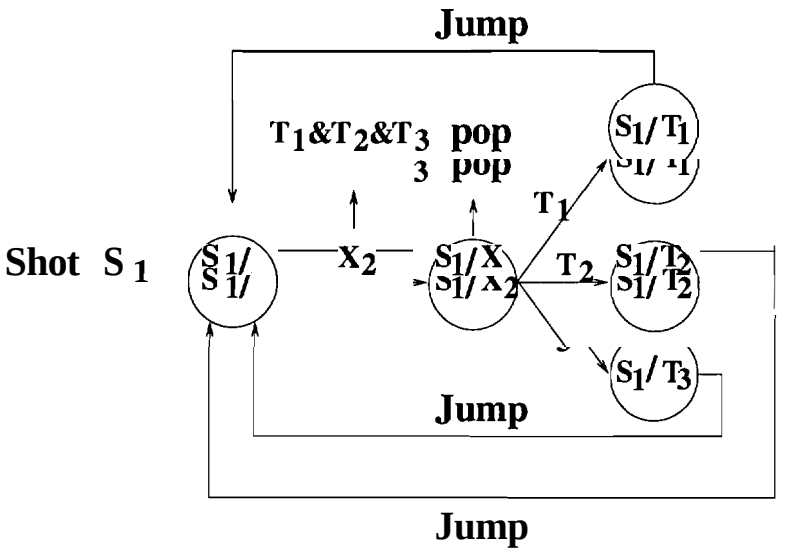


Fig. 6.3. Models loops using *Jump* for shot S_1 as show in Figure 6.2(c).

the subnetworks. When video contents are selected, a part of the multimedia presentation may be invoked so that different media streams will be displayed together with this video media stream. As show in (Chen and Kashyap, 1997a), ATNs can moclel a multimedia presentation with different media streams mixed together. If the bandwidth is not enough for a real-time presentation, a compressed version of media streams can be retrieved instead of full-resolution via the conditions and actions (Chen and Kashyap, 1997b).

Table 6.3 is part of the condition and action table for shot S_1 in Figure 6.2(c). As shown in this table, when in arc 9 input symbol X_2 ($T_1\&T_2\&T_3$) is read of which three selections are provided to users. If users choose T_1 then the input symbol T_1 is read and the control is passed to the subnetwork $T_1/$ in Figure 6.2(d). After subnetwork $T_1/$ returns, in arc 16, the input symbol S_1 is read and the control is passed to the starting state of the same network ($S_1/$) to form the recursion. The condition checked in this case is always true (T) which means no restriction in this situation. Same situations apply when users choose T_2 or T_3 and the control is passed to the subnetworks $T_2/$ and $T_3/$, respectively. Another possibility for modeling user loops is to use *Jump* action to let control go back to a specific state. Figure 6.3 is a subnetwork of an ATN with the same situation as shown in Figure 6.2(c) but using *Jump* instead of recursive calls. In Figure 6.3, there is an arc pointing to the starting state

Table 6.3 "Next-Symbol" is a procedure to read the next input symbol of multimedia input string. "Subnetwork(S_n)" is a procedure to jump to the subnetwork sn.

Arc	Symbol	Condition	Action
9	T_1&T_2&T_3	if choice = T_1	Next_Symbol(T_1) and Subnetwork(T_1/)
		if choice = T_2	Next_Symbol(T_2) and Subnetwork(T_2/)
		if choice = T_3	Next_Symbol(T_3) and Subnetwork(T_3/)
13	S_1	T	Next_Symbol(S_1) and Subnetwork(S_1/)
14	S_1	T	Next_Symbol(S_1) and Subnetwork(S_1/)
15	S_1	T	Next_Symbol(S_1) and Subnetwork(S_1/)

S_1 after states S_1/T_1, S_1/T_2, or S_1/T_3. The multimedia input string for Figure 6.3 is as follows:

$$\text{Multimedia input string: } ((T_1 \& T_2 \& T_3)(T_1|T_2|T_3)^*)^+$$

Input symbol S_1 is no longer in the above multimedia input string if comparing with the multimediit input string for Figure 6.2(c). The benefit to using *Jump* is that we can pass the control to any state in this network which recursive calls cannot do. The disadvantage of this method is that the ATN network becomes complicated when too many arcs point back to the previous states. Moreover, since *Jump* action will let control be passed to the pointing state directly, conditions and actions cannot be imposed when *Jump* arc is used.

6.4 Key frames Selection Based on Temporal and Spatial Analysis of Video Sequences

The next level under shots are key frames. Key frames selections play an important role to let users examine the key changes in each video shot. Since each shot may still have too many video frames, it is reasonable to use key frames to represent the shots. The easiest way of key frame selection is to choose the first frame of the shot. However, this rnethod may miss some important temporal and spatial changes in each shot. The second way is to include all video frames as key frames and this may have computational and storage problems, and may increase users' perception burdens. The third way is to choose key frames based on fixed durations. This method is still not a good mechanism since it may give us many key frames with similar contents. Therefore, how to select key frames to represent a video shot is an important issue for digital library browsing, searching, and retrieval (Yeung and Liu, 1995). To achieve a balance, we propose a key frame selection mechanism based on the number, temporal, and spatial changes of the semantic objects in the video frames. Other features may also be possible for the key frame selections but we focus on the number, temporal, and spatial relations of semantic objects in this study. Therefore, spatio-tem poral changes in each shot can be represented by these key frames. For example, in each shot of a basketball game, players may change positions in subsequent frames and the number of players appearing may change at the time duration of the shot. Let the set of semantic objects in the ith shot (t_i) of the jth frame (f_j) be denoted by O_j^i. We define the key frame selections as follows.

Definition 5.1: Given two contiguous video frames f_a and f_b in ith shot, let the sets of the semantic objects in these two video frames be O_a^i and O_b^i. f_b is a key frame if and only if any of following two conditions is satisfied:

(1) $O_a^i \cap O_b^i \neq O_a^i$

(2) Any semantic object spatial location changes between O_a^i and O_b^i.

Given a video shot s;, let K_i be the set of key frames selected for s_i. Initially the first frame is always selected so $K_i = \{f_1\}$. In addition, let f_{last} be the last selected frame and m be a frame index.

1. Initialize: $f_{last} = f_1$;
   n = 1;
   $K_i = \{f_1\}$;
2. Select f_m for m > n
   if ($(O_m^i \cap O_{last}^i \neq O_m^i)$ OR Spatial_location_change(O_m^i, O_{last}^i)) then
   $f_{last} = f_m$;
   $n = m$;
3. $K_i == K_i \cup f_m$;
   Repeat 2 and **3** until the end of the shot.

The first condition of definition 5.1 models the number of semantic object changes in two contiguous video frames at the same shot. The first part of the if-statement in the above solution algorithm is used to check this situation. The latter part of if-statement checks the second condition of definition 5.1 which is to model the temporal and spatial changes of semantic objects in two contiguous video frames of the shot.

Using the same definition in Table 4.1, one semantic object is chosen to be the target semantic object in each video frame. We adopt the minimal bounding rectangle (MBR) concept in R-tree so that each semantic object is covered by a rectangle. In order to distinguish the relative positions, three dimensional spatial relations are used (as shown in Table 4.1). In this table, twenty-seven numbers are used to distinguish the relative positions of each

semantic object relative to the target semantic object and are represented by subscripted numbers. The centroid point of each semantic object is used for space reasoning so that any semantic abject is mapped to a point object. Therefore, the relative position between the target semantic object and a semantic object can be derived from these centroid points.

In our design, each key frame is represented by an input symbol in a rnultimedia input string. The details of how to use a multimedia input string to represent the temporal and spatial changes of semantic objects can be found in (Chen and Kashyap, 1997b).

Assume Figures 6.4(a), (b), (c), and (d) are four key frames for shot T_1. The multimedia input string to represent these four key frames is as follows:

Multimedia input string: $\underbrace{(A_1\&B_{16}\&C_{25}\&D_{22})}_{R_1}\underbrace{(A_1\&B_{16}\&D_{22})}_{R_2}\underbrace{(A_1\&B_{10}\&D_{22})}_{R_3}\underbrace{(A_1\&B_{13}\&E_{22})}_{R_4}$

As shown in the above multimedia input string, there are four input symbols which are R_1, R_2, R_3, and R_4. Input symbols with smaller numbers appear earlier than those with larger numbers. The "&" symbol between two semantic objects is used to denote that the two semantic objects appear in the same video frame. Figure 6.4(a) is represented by input symbol R_1 where four semantic objects **A, B, C,** and **D** are in the video frame. A_1 indicates that **A** is the target semantic object. B_{16} means **B** is above and to the left of **A,** C_{25} means **C** is above and to the right of **A,** and so on. Figure 6.4(b) modeled by input symbol R_2 in which the semantic object **B** is no longer existing and the other three semantic objects remain at the same locations. In this case, the number of semantic objects is reduced from four to three. This is an example to show how to use multimedia input string to represent number of semantic objects changes. Figure 6.4(c) is represented by input symbol R_3, the semantic object B changes its spatial location from above and left to left of **A.** From this example, the relative spatial relations can be modeled by multimedia input string too. Input symbol R_4 models Figure 6.4(d). In this situation, the semantic object **B** changes to a new spatial location, D disappears, and E appears at the same location as D was in Figure 6.4(c). So, the number associated with B changes from 10 to 13, E is in the input symbol R_4, and D does not exist in R_4. The order of these four key frames is modeled by four input symbols concatenated together to indicate that R_1 appears earlier than R_2 and so on.

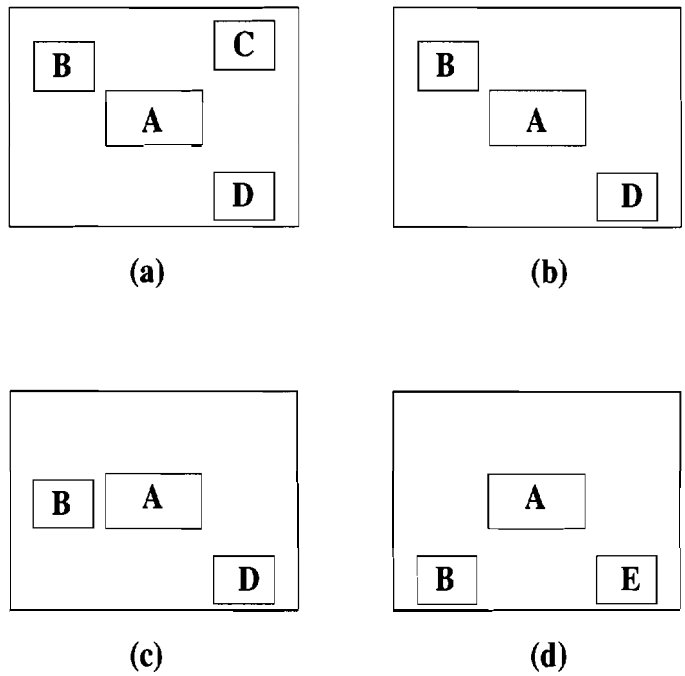


Fig. 6.4. In (a), there are four semantic objects **A, B, C,** and **D** where A is the target semantic object. **C** disappears in (b). The relative position of **B** to **A** changes from above and left to left in (c). In (d), the semantic object D is replaced by E and the semantic object **B** moves to the below and to the left of **A.**

6.5 Sharing Video Units in ATNs

As mentioned in (Sawhney et al., 1997), video units may be shared by different narrative sequences. Some scenes (or shots) may belong to different video clips (or scenes). The production of high-quality multimedia data such as images and video clips takes a great deal of time and is expensive. It is a good way to store these data into large shared databases. Any multimediii application that uses these databases can be greatly facilitated by an underlying model that supports the development process (Schloss and Wynblatt, 1995). The main focus of this underlying model is that the data and the structure can be shared, portable, and reusecl by the designer to create a new multimedia application. Under this design, a multimediii application (presentation) can be separated into several modules based on its conceptual structure. In ATNs, the recursive control mechanisms can be used to model embedded presentations. If an ATN wants to include another existing presentation (ATN) as a subnetwork, the initial state name of the existing presentation (ATN) is put as the arc label of the ATN. This allows any existing presentations to be embedded in the current ATN to make a new design easier. The advantage is that the other presentation structure is independent of the current presentation structure. This makes both the designer and users have a clear view of the presentation. Any change in the shared presentation is done in the shared presentation itself. There is no need to modify those presentations which use it as a subnetwork. This feature allows embedded presentations which can be shared by many presentations and therefore greatly reduces the design time and complexity.

Using the same example as shown in Figure 6.2, assume scene 1 (S_1) and scene 2 (S_2) both contain shot 1 (T_1). Now, shot T_1 is shared by S_1 and S_2. A single subnetwork with the starting state name $T_1/$ as shown in Figure 6.2(d) is shared by S_1 and S_2. Sharing video units among different presentation sequences happens more often in user interactive browsing environments. Different selection paths may go to the same presentation sequence such as vicleo clips, scenes, or shots. Therefore, a semantic model which can model sharing property is very important for designers and users.

6.6 Conclusions

Video data are widely used in today's multimedia applications such as education, video on demand, video conferencing and so on. Managing video data so that users can quickly browse video data is an important issue for the multimedia applications using video data. A good semantic model is needed if we want to meet the needs. In this chapter, ATNs are used to model video hierarchy for browsing. Based on this design, users can view information quickly to decide whether the content is what they want to see. The sharing of video units property lets ATNs include existing media streams, scenes, shots, and presentation sequences very easily. This is useful in distributed multimedia information systems since any browsing environment may include different media elements at different locations. This also provides the designers great flexibility to reuse existing data and presentations in the database or disk storage. Module designs are also possible since data can be shared and reused. Key frames selection based on temporal and spatial relations of semantic objects is used in our design. Under this design, these key frames preserve many of the visual contents and minimize the data size to mitigate the computation and storage problems in multimedia browsing environments. Unlike the existing semantic models which only model presentation, query, or browsing, our ATN model provides these three capabilities into one framework.

7. CONCLUSIONS AND FUTURE WORK

Section 7.1 summarizes the contributions of this work. Then the possible future expansions of this work are discussed in section 7.2.

7.1 Summary of Contributions

As shown in this paper, ATNs together with multimedia input strings can provide three capabilities: presentation, database searching, and browsing. In multimedia database systems and multimedia information systems, modeling the temporal, spatial, or spatio-temporal relations among media streams or semantic objects is very important for both designers and users. The proposed ATN can represent the temporal relations of media streams and spatio-temporal relations of semantic objects easily when associated with multimedia input strings. Experiments to compare the numbers of nodes, arcs, and transitions between ATN and OCPN based on different numbers of media streams at different temporal relations for multimedia presentations are performed in this paper. The results show that ATN needs fewer nodes, arcs, and transitions in all cases. Since ATN needs fewer nodes, arcs, and transitions to represent the multimedia presentation, it is simpler to manage and easier for users to understand. In addition, ATN is a left to right model so that it represents the time flow from left to right. A multimedia transition table in ATN is designed to separate the necessary conditions and actions from the graphical representation. In this regard, users are provided a clear view of the whole structure of the multimedia presentation. Moreover, a graphical representation can be used as the data structure inside the multimedia presentation and the multimedia transition table is the only place that requires memory space. Since each state node allows multiple outgoing arcs, the ATN can. model the user interactions. At the decision point, the multimedia presentation system can display the selection buttons to users for them to make their choices. Different choices invoke different

actions at the multimedia transition table which handles different conditions under different input arc symbols. This feature can free us from the limitations of network delay and jitter, since when a certain situation happens necessary actions can be invoked to handle it. The ATN and its subnetworks can model the hierarchy of a video clip so that they can provide multimediit (video) browsing capability to users. Users can navigate the video contents and choose the segment that they want to watch.

A multimedia input string is the input for the ATN. Since multimedia input strings have the power to express the temporal relations of media streams and spatio-temporal relations of semantic objects, it is a good technique for representing a multimedia presentation and database searching. After the multimedia input string is constructed, a designer can easily modify the input string since symbols are used to represent the media streams. An algorithm to translate the multimedia input string to ATN was proposed in this proposal. Users have the flexibility to use only part of a multimedia input string in the translation algorithm to get a partial ATN graph. Database searching can be a substring matching in the multimedia database queries. Unlike a traditional abstract semantic model that only models the presentation, database searching or browsing, the ATN and its subnetworks can model three of them into one framework.

7.2 Future Work

Future extensions of this work include storage and retrieval of multimedia data, image retrieval based on low-level visual content, human in the loop, mapping of low-level visual feature to high-level concepts, data mining, and new application for Intelligent Transportation Systerns (ITS).

7.2.1 Data Placement and Retrieval for Multimedia Information Systems

In this paper, we do not discuss how to store multimedia data in storage to improve the performance of I/O. Storage layout needs to satisfy the synchronization constraints of media streams and provide smooth real-time presentation to users. Mapping multimedia applications represented as ATN to disk storage in a sequential form needs to be studied so that the latency incurred by seeking can be reduced. Disk-scheduling algorithms and

Admission-control algorithms need to be developed so that disks and buffers can be used efficiently and any stream request can be initiated with successful completion. The disk-scheduling algorithms are used by the server to provide fairness to all client's with reliability and efficiency. The Admission-control algorithms are used to control the initiating of an application and determine what type of service will be provided to this application (Kunii et al., 1995).

7.2.2 Image Retrieval Based on Low-level Visual Content

The number of digital images has grown rapidly. Early on, text or keywords were used to annotate images for image database retrievals such as (Chang and Fu, 1980; Chang et al., 1988). This approach becomes difficult when the number of image collections is large. The second difficulty is different persons or applications may have different perceptions for the same image so that subjectivity and bias may cause unrecoverable mismatches in later retrieval. In order to solve the problem, content-based image retrieval was proposed by using visual content (color, texture, etc.) as indexings (Moni and Kashyap, 1995; Wu, 1997; Gupta et al, 1997). Using feature (content) extraction to get visual features (color, texture, shape, faces, etc.) so that more queries relative to images or video frames can be answered is important. Also, in this study, we assume the low level image processes provide the necessary information for our model such as the minimum bounding box and video segmented into different units. How to segment video sequences into different granularities automatically also needs to be studied.

7.2.3 Hurnan in the Loop

Since "fully automated system" for content-based image retrieval did not lead to success, more recent research focus is on "interactive systems" and "human in the loop". The QBIC team used interactive region segmentation. The MIT team moved from the "automated" Photobook to "interactive" FourEyes (Picard and Minka, 1995; Minka and Picard, 1996). The UCSB team incorporated supervised learning in texture analysis (Ma and Manjunath, 1996). Each party deals with its most suitable parts, thus considerably improving the retrieval performance (Rui et al, 1997(a)(b)). As shown in this paper, the ATN can model

user interactions so that it is possible to include humans in the loop in our model. Further investigation of how to use interactive features in ATN to let users provide feedback in the querying process needs to be conducted.

7.2.4 Lour-level Visual Features and High-level Concepts

In the image and computer vision community, people use image or computer vision techniques to extract low-level features for image retrieval. In real life, hurnans tend to use high-level concepts such as car, tree, and so on. How to link the low-level features to the high level concepts automatically is an important issue for multimedia database queries relative to images or video frames. Therefore, users do not need to first map their high-level information to the low-level features before they issue the queries. In this paper, we assume the semantic objects arc: identified by image processing, computer vision, or manually by people, and do not discuss how to map these two levels. In this study, multimedia input strings consist of media streams or semantic objects. In order to map the low-level visual features to high-level concepts, a possible solution is to model low-level visual features into rnultimedia input strings. Therefore, the relations of low-level and high-level can be connected and both levels of multimedia database queries can be answered.

7.2.5 Data Mining for Multimedia Database Systems

Since data and databases have grown so fast, data mining has gainecl a great deal of attention recently (Fayyad et al., 1996; Silberschatz et al., 1995). Data mining is referred to, or similar to, knowledge discovery in databases, knowledge mining from *databases*, knowledge extraction, data archaelogy, data dredging, data analysis, etc. (Chen et al., 1'996). Many data mining systems have been developed such as QUEST at IBM (Agrawal et al., 1996), KEFIR at the GTE Labs (Matheus et al.,1996), SKICAT at Jet Propulsion Lab, DBMiner at Simon Fraser University (Han et al., 1993), and IMACS at the AT&T Laboratory (Selfridge et al., 1996). Most of these systems work on traditional database systems. Multimedia database systems contain data such as images, video, or audio. How to find previously unknown but useful information from data in multimedia databases becomes very important in many applications. Optimization techniques, very high-level query languages and user interfaces

are needed for decision makers or nonexpert users making queries (Silberschatz et al., 1995). In order to let our ATN model be more powerful, data mining capabilities must be included so that our model can provide more information to users. A possible approach is to put data mining algorithms into condition and action tables to enhance the ability of the ATN.

7.2.6 Multimedia Presentations and Multimedia Database Systems for ITS

Traditional traffic engineering approaches to reducing congestion and improving flow are proving unequal to the growing mobility challenge. Investment in new infrastructure is not a viable solution because of the prohibitively high economic costs, lack of space in urban areas, as well as social and environmental concerns associated with disruption of neighborhoods. In recent years, Intelligent Transportation Systems (ITS), which integrate advances in telecommunications, information systems, automation, and electronics to enhance the efficiency of existing road networks, have emerged as a key approach to addressing the growing mobility problems, and to alleviate congestion and augment the quality of vehicular flow. Distributed computing techniques in the context of dynamic network algorithms for ITS applications and the analysis of the dynamic performance of networks with advanced information systems using distributed dynamic traffic assignment algorithms have been studied by (Chen, 1996; Peeta and Chen, 1997(a)(b)). In order to provide better information to drivers, multimedia technologies which can render information via different medias provide a very good direction to pursue. Our ATN model, which uses subnetworks to model the hierarchy of a video clip, can be applied to the traffic network too. Therefore, how to use ATN to model the network traffic info]-mation will be a good multimedia application so that multimedia presentations and multimedia database searching can be conducted. By using ATN and its subnetworks, the hierarchy relations of zone, link, nodes, and cars can be modeled so that multimedia database queries can be answered and the results can be displayed.

LIST OF REFERENCES

[1] R. Agrawal, M. Mehta, J. Shafer, R. Srikant, A. Arning, and T. Bollinger, "The QUEST Data Mining System," Proc. *Int'l* Conf. Very Large Data Bases, pp. 244-249, Portland, Ore., Aug. 1996.

[2] J.F. Allen, "Maintaining Knowledge About Temporal Intervals," Commun. ACM, Vol. 26, pp. 832-843, Nov. 1983

[3] James Allen, Natural *Language* Understanding, The Benjamin/Cummings Publishing Company, Inc., 1995.

[4] Y.Y. Al-Salqan and C.K. Chang, "Temporal Relations and Synchronization Agents," IEEE Multimedia, pp. 30-39, Summer 1996.

[5] F. Arman, R. Depommer, A. Hsu, and M.Y. Chiu, "Content-based browsing of video sequences," ACM Multimedia 94, pp. 97-103, Aug. 1994.

[6] R. Bayer and E. McCreight, "Organization and Maintenance of Large Ordered Indices," in Proc. 1970 ACM-SIGFIDENT Workshop on Data Description and Access, Houston, Texas, pp. 107-141, Nov. 1970.

[7] A.D. Bimbo, E. Vicario, and D. Zingoni, "Symbolic Description and Visual Querying of Image Sequences Using Spatio-Temporal Logic," IEEE Trans. on *Software* Engineering, vol 7, no. 4, pp. 609-621, August 1995.

[8] G. Blakowski, J. Huebel, and U. Langrehr, "Tools for Specifying and Executing Synchronized Multimedia Presentations," in Proc. 2nd Int'l Workshop on Network and Operating System Support for Digital Audio and Video, pp. 271-279, 1991.

[9] R. Brachman and T. Anand, "The Process of Knowledge Discovery in Databases: A Human-Centered Approach," U.M. Fayyad, G. Piatetsky-Shapiro. P. Smyth, and R. Uthurusamy, eds., Advances in Knowledge Discovery and Data Mining, pp. 37-58, AAAI/MIT Press, 1996.

[10] M. Buchanan and P. Zellweger, "Automatically Generating Consistent Schedules for Multimedia Documents," ACM Multimedia Systems Journal, 1(2), Springer-Verlag, pp. 55-67, 1993.

[11] N.S. Chang and K.S. Fu, "Query-by pictoral-example," IEEE Trans. on Software Engineering, Vol. SE-6, Nov. 1980.

[12] S.K. Chang, C.W. Yan, D.C. Dimitroff, and T. Arndt, "An Intelligent, Image database System," IEEE Trans. *on* Software Engineering, Vol 14, No. 5, pp. 681-688, May 1988.

[13] H.J. Chang, T.Y. Hou, S.K. Chang, "The Management and Application of Teleaction Objects," ACM Multimedia Systems Journal, Vol. 3, pp. 228-237, November 1995.

[14] C.Y. Roger Chen, D.S. Meliksetian, Martin C-S, Chang, L. J. Liu, "Design of a Multimedia Object-Oriented DBMS," ACM Multimedia Systems Journal, Vol. 3, pp. 217-227, Novennber 1995.

[15] Ming-Syan Chen, Jiawei Han, and Philip S. Yu, "Data Mining: An Overview from a Database Perspective," IEEE Trans. on Iinowledge and Data *Engineering*, Vol. 8, No. 6, pp. 866-883, December 1996.

[16] Shu-Ching Chen and R. L. Kashyap, "Temporal and Spatial Semantic Models for Multimedia Presentations," in 1997 International Symposium on Multimedia Information *Processing*, Dec. 11-13, 1997, pp. 441-446.

[17] Shu-Ching Chen and R. L. Kashyap, "A Spatio-Temporal Semantic Model for Multimedia Presentations and Multimedia Database Systems," Submitted to IEEE Trans. on *Knowledge* and Data Engineering.

[18] Shu-Ching Chen and R. L. Kashyap, "Empirical Studies of Multimedia Semantic Models for Multimedia Presentations," in 13th International Conference on *Computer* and Their Applications, Honolulu, Hawaii USA, March 25-27, 1998.

[19] Shu-Ching Chen, "Simulation and Computational Analysis of Distributed Dynamic Network Algorithms for Networks with Advanced Information System," MS Thesis, Purdue University, August, 1996.

[20] Srinivas Peeta and Shu-Ching Chen, "A Distributed Computing Environment for Dynamic Traffic Operations," to appear in Microcomputer Applications in Civil Engineering Journal.

[21] Srinivas Peeta and Shu-Ching Chen, "Analysis of Distributed Computing Techniques for Dynamic Network Algorithms with On-Line ITS Operational Needs," INFORMS, San Deigo meeting, May 4-7 1997.

[22] D. Comer, "The Ubiquitous B-tree," Computing Surveys, 11:2, pp. 121-138, June 1979.

[23] J. Han, Y. Cai, and N. Cercone, "Data-Driven Discovery of Quantitative Rules in Relational Databases," IEEE Trans. *Knowledge* and Data Eng., Vol. 5, pp. 29-40, 1993.

[24] T.L. Kunii, Y. Shinagawa, R.M. Paulg, M.F. Khan, and A.A. Khokhar, "Issues in Storage and Retrieval of Multimedia Data," ACM Multimedia Systems Journal, Vol. 3, pp. 298-304, November 1995.

[25] Y. F. Day, S. Dagtas, M. Iino, A. Khokhar, and A. Ghafoor, "Object-Oriented Concept Modeling of Video Data," March 1995, pp. 401-408, IEEE *Int'l* Conference on Data Engineering, Taipei

[26] Y. F. Day, "Semantic Modeling and Management of Multimedia Data," Ph.D Thesis, Purdue University, August 1996.

[27] B. Falchuk and K. Karmouch, "A multimedia news delivery system over an ATM network," in International conference Multimedia Computing and Systems, 1995, pp. 56-63.

[28] U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, *Advances* in Knowledge *Discovery* and Data Mining, AAAI/MIT Press, 1996.

[29] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, P. Yanker, "Query by Image and Video Content: The QBIC System," IEEE Computer, Vol. 28, No. 9, pp. 23-31, September 1995.

[30] Amarnath Gupta, Simone Santini, and Ramesh Jain, "In Search of Information in Visual Media,," Comm. of the ACM, Vol. 40, No. 12, December 1997, pp. 35-42.

[31] A. Guttman, "R-tree: A Dynamic Index Structure for Spatial Search.," in *Proc.* ACM *SIGMOD*, pp. 47-57, June 1984.

[32] N. Hirzalla, B. Falchuk, and A. Karmouch, "A Temporal Model for Interactive Multimedia Scenarios," IEEE Multimedia, pp. 24-31, Fall 1995.

[33] S.C. Kleene, "Representation of Events in Nerve Nets and Finite Automata, Automata Studies," Princeton University Press, Princeton, N.J., pp. 3-41, 1956.

[34] C.C. Lin, J.X., S.K. Chang, "Transformation and Exchange of Multimedia Objects in Distributed Multimedia Systems," ACM Multimedia Systems Journal, Vol. 4, pp. 12-29, February 1996.

[35] T.D.C. Little and A. Ghafoor, "Synchronization and Storage Models for Multimedia Objects," IEEE J. Selected Areas in *Commun.*, Vol. 9, pp. 413-427, Apr. 1990.

[36] T.D.C. Little and A. Ghafoor, "Interval-Based Conceptual Models for 'Time-Dependent Multimedia Data," IEEE Trans. On Knowledge and Data *Engineering*, Vol. 5, No 4, pp. 551-563, Aug. 1993.

[37] W.Y. Ma and B.S. Manjunath, "Texture features and learning similarity," in *Proc.* IEEE Conf. on Computer Vision and Pattern Recognition, pp. 425-430, 1996.

[38] C.J. Matheus, G. Piatetsky-Shapiro, and D. McNeil, "Selecting and Reporting What is Interesting: The KEFIR Application to Health-Care Data," U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, eds., Advances in *Knowledge* Discovery and Data *Mining*, pp. 495-516, AAAI/MIT Press, 1996.

[39] M. Mills, J. Cohen, and Y.Y. Wong, "A magnifier tool for video data," in *Proc.* ACM Computer Human Interface (CHI), May, 1992, pp. 93-98.

[40] S. Moni and R. L. Kashyap, "A Multiresolution Representation Scheme for Multimedia Databases,?' ACM Multimedia Systems Journal, Vol. 3, pp. 228-237, November 1995.

[41] E. Oomoto, and K. Tanaka, "OVID: Design and Implementation of a Video Object Database System," IEEE Trans. on *Knowledge* and Data Engineering, Vol. 5, No. 4, pp. 629-643, August 1993.

[42] R. Picard and T.P. Minka, "Vision texture for annotation," ACM *Multimedia* Systems Journal: Special Issue on Content-based retrieval, Vol. 3, pp. 3-4, 1995.

[43] T.P. Minka and R.W. Picard, "Interactive learning using a 'society of models'," in Proc. IEEE CVPR, pp. 447-452, 1996.

[44] J.L. Peterson, "Petri nets," ACM Comput. Surveys, Vol. 9, pp. 223-252, Sept. 1977.

[45] Y. Rui, T.S. Huang, and S. Mehrotra, "Content-based image retrieval with relevance feedback in MARS," in Proc. IEEE Int. *Conf.* on Image Proc., 1997

[46] Y. Rui, T.S. Huang, S. Mehrotra, and M. Ortega, "A relevance feedback architecture in content-based multimedia information retrieval systems," in Proc. of IEEE Workshop on Content-based Access of Image and *Video* Libraries, in *conjunction* with IEEE *CVPR'97*, 1997.

[47] Nitin Sawhney, David Balcom, and Ian Smith, "Authoring and Navigating Video in Space and Time," IEEE Multimedia, October-December 1997, pp. 30-39.

[48] G.A. Schloss and M.J. Wynblatt, "Providing definition and temporal structure for multimedia data," *ACM* Multimedia Systems Journal, Vol. 3, pp. 264-277, November 1995.

[49] A. Silberschatz, M. Stonebraker, and J.D. Ullman, "Database Research: Achievements and Opportunities into the 21st Century," Report NSF Workshop *Future* of Database *Systems* Research, May 1995.

[50] S.W. Smoliar and H.J. Zhang, "Content-based video indexing and retrieval," IEEE Multimedia, pp. 62-72, Summer, 1994.

[51] Heiko Thimm and Wolfgang Klas, "δ-Sets for Optimized Relative Adaptive Playout Management in Distributed Multimedia Database Systems," in IEEE 12th International *Conference* on Data Engineering, New Orleans, Louisiana, pp. 584-592, 1996.

[52] W. Woods, "Transition Network Grammars for Natural Language Analysis," Comm. of the ACM, **13,** October 1970, pp. 591-602.

[53] J.-K. Wu, "Content-Based Indexing of Multimedia Databases," IEEE *Trans.* on Knowledge *and* Data Engineering, Vol. 9, No. 6, pp. 978-989, November/December, 1997.

[54] Boon-Lock Yeo and Minerva M. Yeung, "Retrieving and Visualization Video," Comm. of the *ACM*, Vol. 40, No. 12, December 1997, pp. 43-52.

[55] Minerva M. Yeung and Bede Liu, "Efficient Matching and Clustering of Video Shots," in IEEE International Conference on Image Processing, Vol I, October, 1995, pp. 338-341.